

This Page Is Inserted by IFW Operations
and is not a part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

**As rescanning documents *will not* correct images,
please do not report the images to the
Image Problem Mailbox.**



RESEARCH

PRODUCTS

INSIDE DELPHION

[Search](#) | [Advanced](#) | [Boolean](#) | [Proximity](#)
[Search](#) | [Advanced](#) | [Boolean](#) | [Proximity](#) | [Derwent](#)

The Delphion Integrated View: INPADOC Record

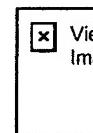
Get Now: PDF | [More choices...](#)Tools: Add to Work File: Create new Work File View: Jump to: [Top](#) Go to: [Derwent](#) [Email this to a friend](#)

>Title: TW0417360B: ERROR CORRECTION DECODING DEVICE FOR EXTENDED REED-SOLOMON CODE, ERROR CORRECTION DEVICE FOR THE FIRST ORDER REED-SOLOMON CODE AND ERROR CORRECTION DEVICE FOR THE SECOND ORDER REED-SOLOMON CODE

Derwent Title: Error correction system for use with extended Reed-Solomon codes - includes circuits using Euclidean algorithms for determining both placement and magnitude of errors prior to their correction [\[Derwent Record\]](#)

Country: TW Taiwan**Kind:** B Patent

Inventor: FUJITA, HACHIRO; Japan
NAKAMURA, TAKAHIKO; Japan
YOSHIDA, HIDEO; Japan



High Resolution

Assignee: MITSUBISHI ELECTRIC CORP Japan
[News, Profiles, Stocks and More about this company](#)

Published / Filed: 2001-01-01 / 1998-02-18**Application Number:** TW1998087102288**IPC Code:** H03M 13/00;**ECLA Code:** None

Priority Number: 1997-04-15 JP1997000097798
1997-08-05 JP1997000211069

Abstract:

The subject of the present invention is that due to the complicated algorithm, it must conduct double calculation based on Euclidean algorithm or Berlekamp-Massey algorithm in the worst case so that the required time for decoding is increased. The method is to generate the syndrome from the received words and estimate the error number generated in the received words; using the estimated error number to change the initial conditions and terminated conditions in the calculation of the Euclidean algorithm so as to derive the error location polynomial and error value polynomial; after conducting Chien's searching to calculate the error location and error value and using the result for error correction; therefore, there only needs one calculation for Euclidean algorithm to proceed decoding.

INPADOC Legal Status: None [Get Now: Family Legal Status Report](#)

Family:

PDF	Publication	Pub. Dat	Filed	Title
<input checked="" type="checkbox"/>	US6131178	2000-10-10	1998-03-04	Error correcting decoding apparatus of extended Reed-Solomon code, and error correcting appara

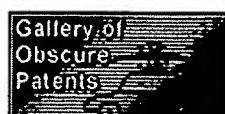
				of singly or doubly extended Reed-Solomon code
<input checked="" type="checkbox"/>	TW0417360B	2001-01-01	1998-02-18	ERROR CORRECTION DECODING DEVICE FOR EXTENDED REED-SOLOMON CODE, ERROR CORRECTION DEVICE FOR THE FIRST ORDER REED-SOLOMON CODE AND ERROR CORRECTION DEVICE FOR THE SECOND ORDER REED-SOLOMON CODE
<input checked="" type="checkbox"/>	JP11003573A2	1999-01-06	1997-08-05	METHOD AND DEVICE FOR ERROR CORRECTING AND DECODING FOR ENLARGED REED-SOLOMON CODE, METHOD AND DEVICE FOR ERROR CORRECTION FOR FIRST ORDER EXTENDED AND ENLARGED REED-SOLOMON CODE, AND METHOD AND DEVICE FOR ERROR CORRECTION FOR SECOND ORDER EXTENDED AND ENLARGED REED-SOLOMON CODE
<input checked="" type="checkbox"/>	FR2762163B1	2001-07-27	1998-03-25	DISPOSITIF DE DECODAGE POUR LA CORRECTION D'ERREURS D'UN CODE REED-SOLOMON ETENDU ET DISPOSITIF DE CORRECTION D'ERREURS DE CODES REED-SOLOMON A SIMPLE OU DOUBLE EXTENSION
<input checked="" type="checkbox"/>	FR2762163A1	1998-10-16	1998-03-25	DISPOSITIF DE DECODAGE POUR LA CORRECTION D'ERREURS D'UN CODE REED-SOLOMON ETENDU ET DISPOSITIF DE CORRECTION D'ERREURS DE CODES REED-SOLOMON A SIMPLE OU DOUBLE EXTENSION
<input checked="" type="checkbox"/>	DE19816347C2	2001-01-25	1998-04-01	Fehlerkorrekturvorrang für einen erweiterten Reed-Solomon-Code, einen einfach erweiterten Reed-Solomon-Code und einen doppelt erweiterten Reed-Solomon-Code
<input checked="" type="checkbox"/>	DE19816347A1	1998-11-05	1998-04-01	Fehlerkorrigierende Decodervorrichtung des erweiterten Reed-Solomon-Codes und fehlerkorrigierende Vorrichtung des einfach oder doppelt erweiterten Reed-Solomon-Codes

7 family members shown above

Other Abstract
Info:



DERABS G1998-545197

[Nominate this for the Gallery...](#)

公告本

417360

417360

申請日期	87 2.18
案 號	87102288
類 別	4-3M 13

A4
C4

(以上各欄由本局填註)

發明專利說明書

一、發明 新型 名稱	中 文	延伸李德所羅門碼的錯誤訂正解碼裝置、一階延伸李德所羅門碼的錯誤訂正裝置以及二階延伸李德所羅門碼的錯誤訂正裝置
	英 文	
二、發明 創作 人	姓 名	1.藤田八郎 2.中村隆彥 3.吉田英夫
	國 籍	日本
三、申請人	住、居所	東京都千代田區丸之內二丁目二番三號
	姓 名 (名稱)	三菱電機股份有限公司
三、申請人	國 籍	日本
	住、居所 (事務所)	東京都千代田區丸之內二丁目二番三號
	代表人 姓 名	北岡隆

四、中文發明摘要(發明之名稱：延伸李德所羅門碼的錯誤訂正解碼
裝置、一階延伸李德所羅門碼的錯
誤訂正裝置以及二階延伸李德所羅
門碼的錯誤訂正裝置)

(請先閱讀發明之注意事項再填寫本頁各欄)

【課題】由於演算法的繁雜，在最差的情況下，必須要進行兩次的歐氏演算法或是 Berlekamp-Massey 演算法之計算，使得解碼所需要的時間變長。

【解決方法】由收訊字組產生微候值，並以此推估在收訊字組中所產生的錯誤個數。利用所推估的錯誤個數，來變更歐氏演算法計算操作時的初始值和終止條件，以求得錯誤位置多項式和錯誤數值多項式。對其進行 Chien 搜尋以計算出錯誤位置和錯誤數值，再根據此來進行錯誤的訂正，藉此只需要進行一次的歐氏演算法計算操作，便可以進行解碼。

英文發明摘要(發明之名稱：)

裝

訂

東

417360

(由本局填寫)	承辦人代碼：
	大類：
	I P C 分類：

A6

B6

本案已向：

國(地區) 申請專利，申請日期：
日本

案號： 有 無 主張優先權
1997/04/15 9-097798
1997/08/05 9-211069

(請先閱讀背面之注意事項再填寫本頁各欄)

裝訂線

有關微生物已寄存於： 寄存日期： 寄存號碼：

五、發明說明(1)

本發明係有關於一種延伸李德所羅門碼(extended Reed-Solomon code，以下稱為延伸 RS 碼)之錯誤訂正解碼裝置，用來對於可以在數位無線通訊和數位磁性記錄中所發生之錯誤加以訂正之延伸 RS 碼，加以解碼；以及可以訂正數位無線通訊和數位磁性記錄中所發生錯誤之一階延伸 RS 碼錯誤訂正裝置和二階延伸 RS 碼錯誤訂正裝置。

首先，針對本發明中做為錯誤訂正解碼之一階延伸 RS 碼之結構加以說明。另外為了簡化起見，高氏場(Galois field)的冪數(characteristic)設為 2。

碼長為 n 並且資料符號(symbol)數為 k 之一階延伸 RS 碼，可利用下述之方式構成。首先，由高氏場之元素(element)所組成之資料符號($c_{n-2}, c_{n-3}, \dots, c_{n-k-1}$)，可以構成下面第(1)式所示之資料多項式。

$$c_{n-2}x^{n-2} + c_{n-3}x^{n-3} + \dots + c_{n-k-1}x^{n-k-1} \quad (1)$$

接著，以 b 為一預定整數，以 α 為構成高氏場原始方程式之根，再將該資料多項式除以下方第(2)式所示之產生(generating)多項式。

$$g(x) = (x - \alpha^b)(x - \alpha^{b+1}) \cdots (x - \alpha^{b+n-k-2}) \quad (2)$$

其結果是以上述第(2)式之除法所得之剩餘多項式，在做為下方所示之第(3)式時，係以添加 $n-k-1$ 符號之係數資料做為檢查符號，產生($c_{n-2}, c_{n-3}, \dots, c_1, c_0$)。

$$c_{n-k-2}x^{n-k-2} + c_{n-k-3}x^{n-k-3} + \dots + c_1x^1 + c_0 \quad (3)$$

接著，以根據下面所示第(4)式所產生之值 c_1 ，添

(請先認真閱讀此之注意事項再填寫本頁)

訂

五、發明說明(2)

加為最後的符號，即構成一階延伸 RS 碼之碼字組

$$C = (c_{n-2}, c_{n-3}, \dots, c_1, c_0, c_{-1})$$

$$c_{-1} = c_{n-2}\alpha^{(b+n-k-1)(n-2)} + c_{n-3}\alpha^{(b+n-k-1)(n-3)} + \dots + c_1\alpha^{(b+n-k-1)} + c_0 \quad (4)$$

在上述所構成之延伸 RS 碼中，已知最小矩離 d 為 $n-k+1$ 。另外，延伸之前的碼 $(c_{n-2}, c_{n-3}, \dots, c_1, c_0)$ 稱之為內部碼，最後添加的一個符號 c_{-1} 則稱為延伸成分。

此處，藉由添加上述第(4)式所產生之值 c_{-1} 做為最後符號所構成之一階延伸 RS 碼的碼字 $C = (c_{n-2}, c_{n-3}, \dots, c_0, c_{-1})$ ，當利用以下第(5)式所設定之 H 做為配類陣列 (parity matrix)時，可以滿足 $H^t C = 0$ 的關係。而 $'C$ 則為 C 的轉置陣列。

$$H = \begin{bmatrix} \alpha^{b(n-2)} & \alpha^{b(n-3)} & \dots & \alpha^b & 1 & 0 \\ \alpha^{(b+1)(n-2)} & \alpha^{(b+1)(n-3)} & \dots & \alpha^{b+1} & 1 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \alpha^{(b+n-k-2)(n-2)} & \alpha^{(b+n-k-2)(n-3)} & \dots & \alpha^{b+n-k-2} & 1 & 0 \\ \alpha^{(b+n-k-1)(n-2)} & \alpha^{(b+n-k-1)(n-3)} & \dots & \alpha^{b+n-k-1} & 1 & 1 \end{bmatrix} \quad (5)$$

在上述所構成之一階延伸 RS 碼中，當最小距離為 d 時，已知 $d=n-k+1$ 。由於內部碼的最小距離為 $n-k$ ，所以延伸的距離僅為 1。當內部碼的最小距離 $n-k$ 為 $2t$ (t 為正整數)時，內部碼僅能訂正 $t-1$ 個錯誤，但是一階延伸 RS 碼的最小距離為 $2t+1$ ，所以能夠利用延伸的特性訂正 t 個錯誤。另外，當內部碼的最小距離 $n-k$ 為 $2t+1$ (t 為正整數)時，此一階延伸 RS 碼的最小距離為 $2t+2$ ，所以能夠訂正 t 個錯誤，以及檢查出 $t+1$ 個錯誤(但是能夠訂正在內部碼中 t 個錯誤以及延伸成分中產生之錯誤，

(請先閱讀背面之注意事項再填寫本頁)

不
訂

五、發明說明(3)

合計為 $t+1$ 個)。藉此，一階延伸 RS 碼是利用在一般 RS 碼中添加上一個檢查符號，來強化其訂正和檢查的能力。

第 30 圖係表示上述碼長 n 、資料符號數 k 、最小距離 $d=n-k+1=2t+1$ (t 為正整數)之一階延伸 RS 碼的習知錯誤訂正方法處理程序之流程圖。圖中，ST60 為徵候值(syndrome)產生步驟，ST61 為歐氏演算法(Euclidean algorithm)計算步驟，ST62 為真偽值判斷步驟，ST63 為歐氏演算法計算步驟，ST64 為真偽值判斷步驟，ST65 為 Chien 搜尋步驟，ST66 為錯誤訂正/檢查步驟。

以下說明其動作。

而在以下之說明中，係以 $R=(r_{n-2}, r_{n-3}, \dots, r_0, r_{-1})$ 做為收訊字組 R 。首先在徵候值產生步驟 ST60 中，係利用下面第(6)式計算徵候值 $S=(S_0, S_1, \dots, S_{2t-2}, S_{2t-1})$ 。而 H 則為上述第(5)式所設的配類陣列。

$$S = H'R \quad (6)$$

接著在歐氏演算法計算步驟 ST61 中，以下面的第(7)式和第(8)式所給的多項式做為初始值，藉由歐氏演算法，進行錯誤位置多項式 $\sigma(x)$ 和錯誤數值多項式 $\omega(x)$ 的計算。

$$R_{-1}(x) = x^{2t-1} \quad (7)$$

$$R_0(x) = S_0 + S_1x + \dots + S_{2t-2}x^{2t-2} \quad (8)$$

而此歐氏演算法計算步驟 ST61 的結束條件則為下面所示之第(9)式。

(請先將背面之注意事項再填寫本頁)

五、發明說明(4)

$$\deg R_i(x) < t - 1$$

(9)

在此歐氏演算法計算步驟 ST61 中對錯誤位置多項式和錯誤數值多項式的計算，則是依照第 31 圖之流程圖順序加以處理。

在此歐氏演算法的計算中，首先分別將 $B_{-1}(x)$ 設定為 0， $B_0(x)$ 設定為 1(步驟 ST610)，再設定上述第(7)式和第(8)做為 $R_{-1}(x)$ 和 $R_0(x)$ 的初始值(步驟 ST611)，並且設定變數 i 的初始值為 1。接著求取以 $R_{i-1}(x)$ 除 $R_{i-2}(x)$ 時之商多項式 $Q(x)$ (步驟 ST613)，再利用此商多項式 $Q(x)$ ，藉由下面的第(10)式和第(11)式計算 $R_i(x)$ 和 $B_i(x)$ (步驟 ST614)。

$$R_i(x) = Q_i(x)R_{i-1}(x) + R_{i-2}(x) \quad (10)$$

$$B_i(x) = Q_i(x)B_{i-1}(x) + B_{i-2}(x) \quad (11)$$

接著判斷藉由此方式所得到的 $R_i(x)$ 的次數是否滿足其終止條件(步驟 ST615)，若不滿足則增加變數 i (步驟 ST616)，並且返回到步驟 ST613 之後的處理。另一方面，如果終止條件滿足，則藉由下面的第(12)式和第(13)式，進行錯誤位置多項式 $\sigma(x)$ 和錯誤數值多項式 $\omega(x)$ 的計算(步驟 ST617)，結束此一歐氏演算法的計算處理。

$$\sigma(x) = B_i(x) \quad (12)$$

$$\omega(x) = R_i(x) \quad (13)$$

接著利用下面的第(14)式，在真偽判斷步驟 ST62 中判斷此歐氏演算法計算步驟 ST61 所計算出的錯誤位置多項式和錯誤數值多項式，是否為真的錯誤位置多項

(請先閱讀背面之注意事項再填寫本頁)

表
訂

五、發明說明(5)

式和錯誤數值多項式。

$$\deg \omega < \deg \sigma \leq t-1 \quad (14)$$

當滿足此條件時，則在 Chien 搜尋步驟 ST65 中對於歐氏演算法計算步驟 ST61 中所計算出的錯誤位置多項式 $\sigma(x)$ 和錯誤數值多項式 $\omega(x)$ ，進行陳氏搜尋，求取錯誤位置以及錯誤值。接著進入錯誤訂正/檢查步驟 ST66，在判斷出此錯誤能夠訂正時進行錯誤訂正的動作，再結束這一連串的處理。

另一方面，當在真偽判斷步驟 ST62 中判斷的結果是不滿足上述條件時，則在歐氏演算法計算步驟 ST63 中以下面第(15)式和第(16)式所示之多項式做為初始值，再回到第 33 圖所示之處理，利用歐氏演算法進行錯誤位置多項式 $\sigma(x)$ 和錯誤數值多項式 $\omega(x)$ 之計算。

$$R_{-1}(x) = x^{2t} \quad (15)$$

$$R_0(x) = S_0 + S_1x + \dots + S_{2t-2}x^{2t-2} + S_{2t-1}x^{2t-1} \quad (16)$$

另外此歐氏演算法計算步驟 ST63 的終止條件則是使用下述之第(17)式。

$$\deg R_i(x) < t \quad (17)$$

接著利用下面的第(18)式，在真偽判斷步驟 ST64 中判斷此歐氏演算法計算步驟 ST63 所計算出的錯誤位置多項式和錯誤數值多項式，是否為真的錯誤位置多項式和錯誤數值多項式。

$$\deg \omega < \deg \sigma \quad (18)$$

當滿足此條件時，則在 Chien 搜尋步驟 ST65 中對

(請先閱讀背面之注意事項再填寫本頁)

表
訂

五、發明說明(6)

於錯誤位置多項式 $\sigma(x)$ 和錯誤數值多項式 $\omega(x)$ ，進行 Chien 搜尋，計算錯誤位置以及錯誤值。如果此錯誤能夠訂正時，則在錯誤訂正/檢查步驟 ST66 中進行錯誤訂正的動作，並且結束這一連串的處理。另一方面，在真偽判斷步驟 ST62 中的判斷結果，如果不能滿足上述條件時，不能夠進行錯誤的訂正，所以錯誤訂正/檢查步驟 ST66 中的處理則僅止於錯誤的檢查，而結束這一連串的處理。

接著對於二階延伸 RS 碼進行說明。另外為了簡化說明，高氏場的幕數設為 2。

碼長 n 且資料符號數為 k 之二階延伸 RS 碼，以下列方式構成。首先，由高氏場之元素所組成之資料符號 $(c_{n-3}, c_{n-4}, \dots, c_{n-k-2})$ 可以構成以下第(19)式所示之資料多項式。

$$c_{n-3}x^{n-3} + c_{n-4}x^{n-4} + \dots + c_{n-k-2}x^{n-k-2} \quad (19)$$

接著，以 b 為一預定整數，以 α 為構成高氏場原始方程式之根，再將該資料多項式除以下面第(20)式所示之產生多項式。

$$g(x) = (x - \alpha)(x - \alpha^{b+1}) \cdots (x - \alpha^{b+k-2}) \quad (20)$$

以此第(20)式之除法所得之剩餘多項式，在做為下方所示之第(21)式時，再添加 $n-k-2$ 符號之係數資料做為檢查符號，產生 $(c_{n-3}, c_{n-4}, \dots, c_1, c_0)$ 。

$$c_{n-k-3}x^{n-k-3} + c_{n-k-4}x^{n-k-4} + \dots + c_1x + c_0 \quad (21)$$

接著，以根據下面所示第(22)式和第(23)式所產生

(請先閱讀背面之注意事項再填寫本頁)

表

訂

五、發明說明(7)

之值 c_{-1} 和 c_{n-2} ，添加做為檢查符號，即構成二階延伸 RS 碼之碼字組 $C=(c_{n-2}, c_{n-3}, \dots, c_0, c_{-1})$ 。

$$c_{-1} = c_{n-3} + c_{n-4} + \dots + c_1 + c_0 \quad (22)$$

$$c_{n-2} = c_{n-3}\alpha^{(n-k-1)(n-3)} + c_{n-4}\alpha^{(n-k-1)(n-4)} + \dots + c_1\alpha^{n-k-1} + c_0 \quad (23)$$

此處，藉由添加上述第(22)式和第(23)式所產生之值 c_{-1} 和 c_{n-2} 做為最後符號所構成之二階延伸 RS 碼的碼字組 $C=(c_{n-2}, c_{n-3}, \dots, c_0, c_{-1})$ ，當利用以下第(24)式所設定之 H 做為配類陣列時，可以滿足 $H_2^t C=0$ 的關係。而 C 則為 C 的轉置陣列。

$$H = \begin{bmatrix} 0 & 1 & 1 & \dots & 1 & 1 & 1 \\ 0 & \alpha^{n-3} & \alpha^{n-2} & \dots & \alpha & 1 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \alpha^{(n-k-2)(n-3)} & \alpha^{(n-k-2)(n-2)} & \dots & \alpha^{n-k-2} & 1 & 0 \\ 1 & \alpha^{(n-k-1)(n-3)} & \alpha^{(n-k-1)(n-2)} & \dots & \alpha^{n-k-1} & 1 & 0 \end{bmatrix} \quad (24)$$

在此場合中，延伸之前的碼 $(c_{n-3}, c_{n-4}, \dots, c_1, c_0)$ 稱為內部碼，添加在最後的符號 c_{-1} 稱為第一延伸成分，添加在最前的符號 c_{n-2} 稱為第二延伸成分。

在上述所構成的二階延伸 RS 碼中，當最小距離為 d 時，已知 $d=n-k+1$ 。由於內部碼的最小距離為 $n-k-1$ ，所以延伸的距離僅為 2。因此能夠對內部碼的訂正能力，增加訂正一個錯誤的能力。

對於習知二階延伸 RS 碼的錯誤訂正能力，已知者如 R.E.Blahut 所著之 "Theory and Practice of Error Control Codes" (Addison Wesley, 1984) 第 9 章第 3 節所述之演算法。

(請先閱讀背面之注意事項再填寫本頁)

訂

五、發明說明(8)

第32圖係表示上述碼長 n 、資料符號數 k 、最小距離 $d=n-k+1=2t+1$ (t 為正整數)之二階延伸RS碼的習知錯誤訂正方法處理程序之流程圖。

圖中，ST70為徵候值產生步驟，ST71為Berlekamp-Messey演算法計算步驟，ST72為錯誤位置多項式的測試步驟，ST73為Berlekamp-Messay演算法計算步驟，ST74為錯誤位置多項式的測試步驟，ST75為Berlekamp-Messey演算法計算步驟，ST76為錯誤位置多項式的測試步驟，ST77為Berlekamp-Messey演算法計算步驟，ST78為錯誤位置多項式的測試步驟，ST79為Chien搜尋步驟，ST80為根數量測試步驟，ST81為錯誤訂正步驟。

以下說明其動作。

而在以下之說明中，係以 $R=(r_{n-2}, r_{n-3}, \dots, r_0, r_1)$ 做為收訊字組 R 。首先在徵候值產生步驟 ST70 中，係利用下面第(25)式計算徵候值 $S=(S_0, S_1, \dots, S_{2t-2}, S_{2t-1})$ 。此處包含第一延伸成分之錯誤資料者為 S_0 ，包含第二延伸成分之錯誤資料者為 S_{2t-1} 。

$$S = H_2' R \quad (25)$$

接著在 Berlekamp-Massey 演算法計算步驟 ST71 中，利用僅含內部碼所產生之錯誤資料的 $2t-2$ 個徵候值 S_1, \dots, S_{2t-2} ，進行 Berlekamp-Massey 演算法(BM1)。第33圖表示用來計算錯誤位置多項式之 Berlekamp-Massey 演算法計算程序的流程圖。

(請先閱讀背面之注意事項再填寫本頁)

訂

五、發明說明(9)

以下參考第33圖，說明此Berlekamp-Massey演算法(BM1)。

首先，將錯誤位置多項式的初始值 $\sigma^{(0)}(x)$ 設為1，形式移位暫存器的長度L設為0，用來更新錯誤位置多項式之輔助多項式 $\tau^{(0)}(x)$ 設為1，用來計數步驟數量的r設為1(步驟ST710)。

逐次地計算錯誤位置多項式，直到第r-1次步驟結束後，計算出錯誤位置多項式 $\sigma^{(r-1)}(x)$ 和輔助多項式 $\tau^{(r-1)}(x)$ 。

在第r次步驟中，利用 $\sigma^{(r-1)}(x)$ 的係數，計算下面所示第(26)式中稱為偏差(discrepancy)的值 Δ_r (步驟ST711)。

$$\Delta_r = \sum_{j=0}^l \sigma_j^{(r-1)} S_{r-j} \quad (26)$$

接著檢查此計算結果(步驟ST712)，若偏差值 Δ_r 為0時，則不更新錯誤位置多項式，並且以第r-1次所得之 $\sigma^{(r-1)}(x)$ 做為第r次的錯誤位置多項式(步驟ST713)。另一方面，若偏差值 Δ_r 不為0時，則使用輔助多項式 $\tau^{(r-1)}(x)$ ，以下面第(27)式構成接續多項式 $\eta(x)$ (步驟ST714)。

$$\eta(x) \leftarrow \sigma^{(r-1)}(x) + \Delta_r x \tau^{(r-1)}(x) \quad (27)$$

接著檢查形式暫存器長度L是否滿足 $2L < r$ (步驟ST715)，若滿足時，錯誤位置多項式 $\sigma^{(r)}(x)$ ，輔助多項式 $\tau^{(r)}(x)$ 以及形式暫存器長度L則以第(28)式~第(30)式所示之方式加以更新(步驟ST716)。

$$\tau^{(r)}(x) \leftarrow \Delta_r^{-1} \sigma^{(r-1)}(x) \quad (28)$$

$$\sigma^{(r)}(x) \leftarrow \eta(x) \quad (29)$$

(請先閱讀背面之注意事項再填寫本頁)

表

訂

五、發明說明 (10)

$$L \leftarrow r - L \quad (30)$$

另外，當此不等式不滿足時，形式暫存器長度 L 不做更新，錯誤位置多項式 $\sigma^{(r)}(x)$ 則以下面第(31)式的方式更新(步驟 ST717)，而輔助多項式 $\tau^{(r)}(x)$ 則以下面第(32)式的方式更新(步驟 ST713)。

$$\sigma^{(r)}(x) \leftarrow \eta(x) \quad (31)$$

$$\tau^{(r)}(x) \leftarrow x\tau^{(r-1)}(x) \quad (32)$$

以下則在檢查出用來計數步驟數量之變數 r 達到 $2t-2$ 之前(步驟 ST718)，一邊增加變數 r (步驟 ST719)，一邊返回步驟 ST711 之後的處理。

當此 Berlekamp-Massey 演算法計算步驟 ST71 中的 Berlekamp-Massey 演算法(BM1)處理結束時，在測試步驟 ST72 中，則是進行錯誤位置多項式 $\sigma(x)$ 是否適當的測試。此處，在內部碼產生 $t-2$ 個以下錯誤時，第 $2t-3$ 次和第 $2t-2$ 次偏差值為 0，而在 Berlekamp-Massey 演算法(BM1)結束後，形式暫存器長度 L 在 $t-2$ 以下，錯誤位置多項式 $\sigma^{(2t-2)}(x)$ 的階數則等於形式暫存器長度 L 。因此，當錯誤位置多項式 $\sigma^{(2t-2)}(x)$ 滿足下面第(33)式所示之條件時，則判斷錯誤位置多項式 $\sigma^{(2t-2)}(x)$ 為適當的，並進入 Chien 搜尋步驟 ST79，實施 $\sigma(x)=\sigma^{(2t-2)}(x)$ 中的 Chien 搜尋。

$$\Delta_{2t-3} = 0 \quad \text{並且}$$

$$\Delta_{2t-2} = 0 \quad \text{並且}$$

$$L \leq t-2 \quad \text{並且}$$

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

五、發明說明(11)

$$\deg \sigma^{(2t-2)} = L \quad (33)$$

另一方面，當上述第(33)式所示之條件不成立時，推定在內部碼中產生 $t-1$ 個以上的錯誤碼，而在 Berlekamp-Massey 演算法計算步驟 ST73 中，使用 S_{2t-1} 再續繼進行一個步驟的 Berlekamp-Massey 演算法(BM1')。接著在測試步驟 ST74 中測試錯誤位置多項式 $\sigma(x)$ 是否適當。在內部碼中產生 $t-1$ 個錯誤並且在第二延伸成分未產生錯誤時，偏差值 Δ_{2t-1} 則為 0。另外，錯誤位置多項式 $\sigma^{(2t-1)}(x)(=\sigma^{(2t-2)}(x))$ 的階數則等於形式暫存器長度 L 。因此，當錯誤位置多項式 $\sigma^{(2t-1)}(x)$ 滿足下面第(34)式所示之條件時，則判斷錯誤位置多項式 $\sigma^{(2t-1)}(x)$ 為適當的，並進入 Chien 搜尋步驟 ST79，實施 $\sigma(x)=\sigma^{(2t-1)}(x)$ 中的 Chien 搜尋。

$$\Delta_{2t-1} = 0 \text{ 並且}$$

$$\deg \sigma^{(2t-1)} = L \quad (34)$$

當上述第(34)式所示之條件不成立時，推定在內部碼中產生 t 個以上的錯誤碼，而在 Berlekamp-Massey 演算法計算步驟 ST75 中，使用 $S_0, S_1, \dots, S_{2t-2}, S_{2t-1}$ 再度進行 Berlekamp-Massey 演算法(BM2)。基本上此 Berlekamp-Massey 演算法(BM2)與 Berlekamp-Massey 演算法(BM1)相同，除了偏差值的計算係以下面所示之第(35)式進行。

$$\Delta_r = \sum_{j=0}^L \sigma_j^{(r-1)} S_{r-j-1} \quad (35)$$

(請先閱讀背面之注意事項再填寫本頁)

裝訂

五、發明說明 (12)

接著在測試步驟 ST76 中，進行錯誤位置多項式 $\sigma(x)$ 是否適當的測試。此處，在內部碼中產生 $t-1$ 個錯誤並且在第一延伸成分中未產生錯誤的情況下，偏差值 Δ_{2t-1} 則為 0。另外，在 Berlekamp-Massey(BM2) 結束後，錯誤位置多項式 $\sigma^{(2t-1)}(x)(=\sigma^{(2t-2)}(x))$ 的階數等於形式暫存器長度 L。因此，當錯誤位置多項式 $\sigma^{(2t-1)}(x)$ 滿足第(34)式之條件時，則判斷錯誤位置多項式 $\sigma^{(2t-1)}(x)$ 為適當的，並且進入 Chien 搜尋步驟 ST79，實施 $\sigma(x)=\sigma^{(2t-1)}(x)$ 中的 Chien 搜尋。

另外，當第(34)式的條件不成立時，推定在內部碼中產生 t 個以上的錯誤碼，而在 Berlekamp-Massey 演算法計算步驟 ST77 中，使用 S_{2t-1} 再續繼進行一個步驟的 Berlekamp-Massey 演算法(BM2')。若在兩個延伸成分中未產生錯誤時，當 Berlekamp-Massey(BM2') 結束後，錯誤位置多項式 $\sigma^{(2t)}(x)$ 的階數等於形式暫存器長度 L。因此，當下面第(36)式的條件成立時，則判斷錯誤位置多項式 $\sigma^{(2t)}(x)$ 為適當的，並且進入 Chien 搜尋步驟 ST79，實施 $\sigma(x)=\sigma^{(2t)}(x)$ 中的 Chien 搜尋。

$$\deg \sigma^{(2t)} = L = t \quad (36)$$

另一方面，當第(36)式的條件不成立時，即推定產生 $t+1$ 個以上的錯誤，無法進行錯誤訂正，所以在送出錯誤檢查旗標時一併送出原來的收訊字組 R。

在上述 Berlekamp-Massey 演算法計算步驟 ST71、73、75、77 中能夠測試出錯誤位置多項式的情況下，則

(請先閱讀背面之注意事項再填寫本頁)

表

訂

五、發明說明 (13)

在 Chien 搜尋步驟 ST79 中進行 Chien 搜尋，並且進行錯誤位置多項式 $\sigma(x)$ 的根(錯誤位置)之計算。接著在測試步驟 ST80 中，則判斷所得到的錯誤位置多項式 $\sigma(x)$ 的根數量是否與此錯誤位置多項式 $\sigma(x)$ 的階數一致。一致時則在錯誤訂正步驟 ST81 中計算錯誤數值，並且從收訊字組 R 錯誤位置的符號中減去錯誤數值，進行錯誤的訂正。

另外，若所得到錯誤位置多項式 $\sigma(x)$ 的根數量與錯誤位置多項式 $\sigma(x)$ 的階數不一致時，無法進行錯誤訂正，所以在送出錯誤檢查旗標時一併送出原來的收訊字組 R。

由於習知一階延伸 RS 碼的錯誤訂正係以上述的方式進行，所以演算法相當繁複，在最差的情況下必須要進行兩次的歐氏演算法的計算，於是便造成解碼所需時間過長的問題。另外，尚無有關於此一階延伸 RS 碼在消失(erasure)/錯誤訂正上的公開文獻，因此有必要開發出有效率的消失/錯誤訂正方法和裝置。

另外，由於習知二階延伸 RS 碼的錯誤訂正係以上述的方式進行，所以演算法相當繁複，在最差的情況下必須要進行兩次的 Berlekamp-Massey 演算法的計算，於是便造成解碼所需時間過長的問題。另外，尚無有關於此二階延伸 RS 碼在消失(erasure)/錯誤訂正上的公開文獻，因此有必要開發出有效率的消失/錯誤訂正方法和裝置。

(請先閱讀背面之注意事項再填寫本頁)

裝

訂

五、發明說明 (14)

為了解決上述之問題，本發明之目的是在於提供一種錯誤訂正解碼裝置，單單進行一次歐氏演算法的計算，即能夠求出錯誤位置多項式以及錯誤數值多項式。

本發明之另一目的在於提供一種錯誤訂正解碼裝置，即使在延伸成本中發生錯誤的場合下，也能夠計算錯誤的數值。

本發明之又一目的在於提供一種錯誤訂正解碼裝置，能夠減少解碼時所需要的步驟數量。

本發明之又一目的在於提供一種一階延伸 RS 碼之錯誤訂正裝置，單單進行一次歐氏演算法的計算，即能夠求出錯誤位置多項式以及錯誤數值多項式。

本發明之又一目的在於提供一種一階延伸 RS 碼之錯誤訂正裝置，能夠有效率地進行一階延伸 RS 碼的消失/錯誤訂正。

本發明之又一目的在於提供一種二階延伸 RS 碼之錯誤訂正裝置，單單進行一次 Berlekamp-Massey 演算法的計算，即能夠求出錯誤位置多項式。

本發明之又一目的在於提供一種二階延伸 RS 碼之錯誤訂正裝置，能夠有效率地進行二階延伸 RS 碼的消失/錯誤訂正。

本發明之又一目的在於提供一種二階延伸 RS 碼之錯誤訂正裝置，單單進行一次 Berlekamp-Massey 演算法的計算，即能夠求出錯誤位置多項式以及錯誤數值多項式。

(請先閱讀背面之注意事項再填寫本頁)

表

訂

五、發明說明 (15)

第一，本發明中所提供的延伸李德-所羅門碼之錯誤訂正解碼裝置，其包括：徵候值產生裝置，用以根據所輸入之收訊字組，計算出其徵候值；收訊字組儲存裝置，用以儲存上述收訊字組；錯誤個數估算裝置，根據上述徵候值產生裝置所計算出的徵候值，估算錯誤個數；歐氏演算法計算裝置，其對應於上述錯誤個數估算裝置所估算出之錯誤個數，設定用以計算錯誤位置多項式和錯誤數值多項式之初始值和終止條件，並且藉由歐氏演算法，計算錯誤位置多項式和錯誤數值多項式；Chien搜尋裝置，根據上述歐氏演算法計算裝置所計算出之錯誤位置多項式和錯誤數值多項式，計算錯誤位置和錯誤數值；以及錯誤訂正裝置，根據上述 Chien 搜尋裝置的計算結果，對儲存於上述收訊字組儲存裝置中之收訊字組，訂正其錯誤。

此處，上述延伸李德-所羅門碼之錯誤訂正解碼裝置，亦可以藉由具有高氏場上加法裝置和乘法裝置之演算裝置所構成，其中上述 Chien 搜尋裝置設置有徵候值修正裝置，係藉由基於上述歐氏演算法計算裝置所計算出的錯誤位置多項式和錯誤數值多項式，所算出之錯誤位置和錯誤數值，以修正收訊字組徵候值的方式，計算出延伸成分中的錯誤數值。

另外，上述錯誤個數估算裝置亦可以由高氏場上乘法裝置、加法裝置和儲存裝置所構成，對於最小距離為 d 之延伸李德-所羅門碼，係根據在上述徵候值產生裝置

(請先閱讀背面之注意事項再填寫本頁)

訂

五、發明說明 (16)

中所計算之收訊字組的徵候值，判斷是否有錯誤，在存在錯誤時，則判斷其個數低於 $[(d-1)/2]$ 個或是等於 $[(d-1)/2]$ 個。而歐氏演算法計算裝置係在上述錯誤個數估算裝置判斷出存在錯誤時，根據錯誤個數是低於 $[(d-1)/2]$ 個或是等於 $[(d-1)/2]$ 個之情況，進行用以計算錯誤位置多項式和錯誤數值多項式之初始值和終止條件的設定，藉由歐氏演算法，計算錯誤位置多項式和錯誤數值多項式。

另外，上述錯誤個數估算裝置對於最小距離為 d 之延伸李德-所羅門碼，係根據在上述徵候值產生裝置中所計算之收訊字組的徵候值，判斷是否有錯誤，在存在錯誤時，則判斷其個數低於 $[(d-1)/2]$ 個、等於 $[(d-1)/2]$ 個或高於 $[(d-1)/2]$ 個。上述歐氏演算法計算裝置在上述錯誤個數估算裝置判斷出錯誤個數高於 $[(d-1)/2]$ 個時，中止藉由歐氏演算法對錯誤位置多項式和錯誤數值多項式之計算。上述 Chien 搜尋裝置在上述錯誤個數估算裝置判斷出錯誤個數高於 $[(d-1)/2]$ 個時，中止藉由 Chien 搜尋對錯誤位置和錯誤數值之計算。上述錯誤訂正裝置在上述錯誤個數估算裝置判斷出錯誤個數高於 $[(d-1)/2]$ 個時，輸出儲存在上述收訊字組儲存裝置中之原來收訊字組。

第二，本發明中所提供之階延伸李德-所羅門碼之錯誤訂正裝置，其包括：徵候值產生裝置，用以根據所輸入之收訊字組，計算出其徵候值；收訊字組儲存裝置，用以儲存上述收訊字組；徵候值多項式產生裝置，

(請先閱請背面之注意事項再填寫本頁)

表
訂

五、發明說明 (17)

係對上述徵候值產生裝置所計算出者之中包含延伸成分錯誤資訊的徵候值設為常數項之方式，產生徵候值多項式；歐氏演算法計算裝置，用以將上述徵候值多項式產生裝置所產生之徵候值多項式，設為初始值，並且藉由歐氏演算法，計算錯誤位置多項式和錯誤數值多項式；Chien 搜尋裝置，根據上述歐氏演算法計算裝置所計算出之錯誤位置多項式和錯誤數值多項式，計算錯誤位置和錯誤數值；以及錯誤訂正裝置，根據上述 Chien 搜尋裝置所計算之錯誤位置和錯誤數值，用以訂正儲存於上述收訊字組儲存裝置中之收訊字組的錯誤。

此處，上述之一階延伸李德-所羅門碼的錯誤訂正裝置，更包括消失位置係數產生裝置，其藉由伴隨於上述收訊字組之消失旗標，產生消失位置係數；消失位置多項式產生裝置，其根據上述消失位置係數產生裝置所產生之消失位置係數，產生消失位置多項式；以及修正徵候值多項式產生裝置，用以將上述徵候值多項式產生裝置所產生之徵候值多項式和上述消失位置多項式產生裝置所產生之消失位置多項式相乘，產生修正徵候值多項式。其中上述歐氏演算法計算裝置係將上述修正徵候值多項式產生裝置所產生之修正徵候值多項式，設為初始值，並藉由歐氏演算法計算錯誤消失位置多項式和錯誤消失數值多項式。上述 Chien 搜尋裝置係根據上述歐氏演算法計算裝置所算出之錯誤消失位置多項式和錯誤消失數值多項式，計算錯誤位置和錯誤數值。

(請先閱讀背面之注意事項再填寫本頁)

表

訂

五、發明說明 (18)

第三，本發明中所提供的二階延伸李德-所羅門碼之錯誤訂正裝置，其包括：徵候值產生裝置，用以根據所輸入之收訊字組，計算出其徵候值；收訊字組儲存裝置，用以儲存上述收訊字組；Berlekamp-Massey 演算法計算裝置，其採用上述徵候值產生裝置所計算之徵候值，進行 Berlekamp-Massey 演算法，並藉由形式移位暫存器長度值，控制是否使用包含第二延伸成分錯誤資訊之徵候值，以依序計算錯誤位置多項式；Chien 搜尋裝置，對於上述 Berlekamp-Massey 演算法計算裝置所計算出之錯誤位置多項式，進行 Chien 搜尋，計算錯誤位置；錯誤數值產生裝置，當上述 Chien 搜尋裝置所計算之錯誤位置數量適當時，利用基於上述徵候值產生裝置所計算之徵候值而產生之徵候值多項式，計算錯誤數值；以及錯誤訂正裝置，根據上述 Chien 搜尋裝置所計算之錯誤位置和上述錯誤數值產生裝置所計算出之錯誤數值，用以訂正儲存於上述收訊字組儲存裝置中之收訊字組的錯誤。

此處，上述之二階延伸李德-所羅門碼的錯誤訂正裝置，更包括：消失位置係數產生裝置，其藉由伴隨於上述收訊字組之消失旗標，產生消失位置係數；以及消失位置多項式產生裝置，其根據上述消失位置係數產生裝置所產生之消失位置係數，產生消失位置多項式。其中上述 Berlekamp-Massey 演算法計算裝置係將上述消失位置多項式產生裝置所產生之消失位置多項式，設為初

(請先閱讀背面之注意事項再填寫本頁)

裝
訂

五、發明說明(19)

始值，並且藉由形式移位暫存器長度值，控制是否使用包含第二延伸成分之錯誤資訊的徵候值，以依序計算錯誤位置多項式。上述 Chien 搜尋裝置係對於上述 Berlekamp-Massey 演算法計算裝置所算出之錯誤位置多項式，進行 Chien 搜尋。

較佳實施例：

以下說明本發明的實施例。

【第一實施例】

第 1 圖表示本發明第一實施例中延伸 RS 編之錯誤訂正解碼方法處理程序之流程圖。圖中，ST11 是從收訊字組進行徵候值演算的徵候值產生步驟，ST12 是從徵候值產生步驟 ST11 中所算出的徵候值，用來估算在收訊字組中延伸成分以外部分之錯誤個數的錯誤個數估算步驟。ST13 則是基於對應錯誤個數估算步驟 ST12 所估算之錯誤個數而設定的初始值和終止條件，藉由歐氏演算法計算錯誤位置多項式和錯誤數值多項式之歐氏演算法計算步驟。ST14 則是對應於歐氏演算法計算步驟 ST13 所得到之錯誤位置多項式和錯誤數值多項式，進行 Chien 搜尋，用以計算錯誤位置和錯誤數值之 Chien 搜尋步驟。ST15 則是根據 Chien 搜尋步驟 ST14 所算出的錯誤位置和錯誤數值，用來訂正收訊字組中錯誤的錯誤訂正步驟。

另外，第 2 圖~第 6 圖則分別詳細表示上述各步驟

(請先閱讀背面之注意事項再填寫本頁)

校
訂

五、發明說明 (20)

中處理細節的流程圖，第 2 圖為徵候值產生步驟 ST11 的處理程序，第 3 圖為錯誤個數估算步驟 ST12 的處理程序，第 4 圖為歐氏演算法計算步驟 ST13 的處理程序，第 5 圖為 Chien 搜尋步驟 ST14 的處理程序，第 6 圖為錯誤訂正步驟 ST15 的處理程序。

接著說明其動作。

在以下的說明，是在習知延伸 RS 碼的解碼動作中所說明的碼中， $b=0$ 並且最小距離為 7 的延伸 RS 碼來做說明，亦即檢查符號數為 6 個符號的碼。首先，在徵候值產生步驟 S11 中，利用以下的第(37)式和第(38)式，以收訊字組進行徵候值 $S_i(i=0, 1, \dots, 5)$ 的計算。

$$S_i = \sum_{j=0}^{n-2} r_j \alpha^{(i+1)j} \quad (i=0, 1, \dots, 4) \quad (37)$$

$$S_5 = r_{-1} + \sum_{j=0}^{n-2} r_j \alpha^{6j} \quad (38)$$

亦即，如第 2 圖所示，首先徵候值 $S_i(i=0, 1, \dots, 5)$ 的初始值分別被設定為 0，同時將旗標設為”關”($F=0$)(步驟 ST110)。接著將變數 j 的初始值設為 $n-2$ (步驟 ST111)，此時的 $S_i(i=0, 1, \dots, 5)$ 則依下面第(39)式所示般，根據在此之前 S_i 值來計算(步驟 ST112)。

$$S_i = r_i + S_i \alpha^{i+1} \quad (39)$$

此處， α 為構成高氏場之原始方程式的根。

接著，判斷變數 j 是否為 0(步驟 ST113)，若不是則遞減變數 j (步驟 ST114)，並且返回步驟 ST112 的計算更新徵候值 $S_i(i=0, 1, \dots, 5)$ ，一直到變數 j 等於 0 為止。

(請先閱讀背面之注意事項再填寫本頁)

表一
訂

五、發明說明 (21)

當變數 j 為 0 時，則僅對於徵候值 S_5 ，將收訊字組延伸成分的符號 r_1 加在此時的 S_5 值，求得新的徵候值 S_5 (步驟 ST115)。藉此，在此徵候值產生步驟 ST11 中，即結束利用上述第(37)式和第(38)式計算徵候值 S_i 的處理。

接著，在錯誤個數估算步驟 ST12 中，則是由上述徵候值產生步驟 ST11 所推算出之徵候值 $S_i(i=0, 1, \dots, 5)$ ，進行延伸成分以外之錯誤個數 e 的判定。第 3 圖則表示此錯誤個數判定之處理程序流程圖之一例。如第 3 圖所示，在判定錯誤個數時，首先是判斷上述徵候值產生步驟 ST11 中所計算出的徵候值 S_i 是否全部為 0(步驟 ST120)。若全部為 0 時則推測沒有錯誤($e=0$)(步驟 ST121)，並且結束此錯誤個數估算步驟 ST12 的處理。另一方面，若判斷出有錯誤發生時，則以收訊字組的徵候值 S_i ，利用下面的第(40)式~第(42)式來計算 $T_1 \sim T_3$ (步驟 ST122)，再利用第(43)式計算 U 的值(步驟 ST123)，分別以這些值來進行錯誤個數的估數。

$$T_1 = S_1^2 + S_0S_2 \quad (40)$$

$$T_2 = S_2^2 + S_0S_4 \quad (41)$$

$$T_3 = S_3^2 + S_2S_4 \quad (42)$$

$$U = S_4T_1 + S_2T_2 + S_0T_3 \quad (43)$$

接著則是判斷 $T_1=T_2=T_3=0$ 是否成立(步驟 ST124)，其成立時則估算錯誤個數 e 為 1 個(步驟 ST125)，並結束此處理。另外，若 $T_1=T_2=T_3=0$ 不成立時，則進行 U 值是否為 0 的判斷(步驟 ST126)。當 $U=0$ 時則估算錯誤

(請先閱讀背面之注意事項再填寫本頁)

訂

五、發明說明 (22)

個數 e 為 2 個(步驟 ST127)，而若 U 為 0 以外其他值時，則估算錯誤個數 e 為 3 個(步驟 ST128)，並結束此處理。

接著在歐氏演算法計算步驟 ST13 中，則是根據上述錯誤個數估算步驟 ST12 所估算的錯誤個數值 e ，變更初始值和最終條件，並且利用歐氏演算法進行錯誤位置多項式 $\sigma(z)$ 和錯誤數值多項式 $\omega(z)$ 的計算。第 4 圖所示之流程圖，則是表示此歐氏演算法計算步驟 ST13 的詳細處理程序。當歐氏演算法計算步驟 ST13 開始時，首先判斷在錯誤個數估算步驟 ST12 中所估算的錯誤個數 e 是否為 0(步驟 ST130)。若所估算的錯誤個數 e 為 0 個時，則結束此一歐氏演算法計算的處理，並且輸出原來的收訊字組。

另一方面，當有錯誤發生時(所估算的錯誤個數 e 為 0 以外的情況)，則進行歐氏演算法初始值的設定(步驟 ST131)。此處，在藉由此步驟 ST131 所設定的初始值中，用來計算錯誤位置多項式 $\sigma(z)$ 的多項式 $R_i(z)$ 之初始值，是藉由下面的第(44)式和第(45)式，對應於錯誤個數 e 來加以設定；而用來計算錯誤數值多項式 $\omega(z)$ 之多項式 $B_i(z)$ 的初始值，則分別被設定成 $B_{-1}(z)=0$ 和 $B_0(z)=1$ 。

$$R_{-1}(z) = z^{2e} \quad (44)$$

$$R_0(z) = S_{2e-1}z^{2e-1} + S_{2e-2}z^{2e-2} + \dots + S_1z + S_0 \quad (45)$$

接著在初始化變數 i 為 1 之後(步驟 ST132)，則求取多項式 $R_{i-1}(z)$ 除以 $R_{i-2}(z)$ 時的商多項式 $Q(z)$ (步驟 ST133)。並習知延伸 RS 碼的錯誤訂正解碼情況一樣，

(請先閱讀背面之注意事項再填寫本頁)

訂

五、發明說明 (23)

利用所求得的商多項式 $Q(z)$ ，進行餘式多項式 $R_i(z)$ 和 $B_i(z)$ 的計算(步驟 ST134)。接著，判斷上述所得之餘式多項式 $R_i(z)$ 的階數是否小於錯誤個數 e 的值，亦即是否滿足下面第(46)式所示之終止條件(步驟 ST135)。

$$\deg R_i(z) < e \quad (46)$$

結果若判斷出不滿足上述第(46)式的終止條件時，亦即餘式多項式 $R_i(z)$ 的階數尚比錯誤個數 e 來得大時，遞增變數 i (步驟 ST136)，並且回到上述步驟 ST133 和步驟 ST134 的計算。另一方面，若判斷步驟 ST135 中的終止條件可以滿足時，亦即餘式多項式 $R_i(z)$ 的階數比錯誤個數 e 來得小時，則結束上述處理，並且與習知延伸 RS 碼的錯誤訂正解碼情況一樣，以所得到的餘式多項式 $R_i(z)$ 和 $B_i(z)$ 來計算錯誤位置多項式 $\sigma(z)$ 和錯誤數值多項式 $\omega(z)$ (步驟 ST137)，再結束此藉由歐氏演算法所做的計算處理。

接著在 Chien 搜尋步驟 ST14 中，對於在上述歐氏演算法計算步驟 ST13 所計算出的錯誤位置多項式 $\sigma(z)$ 和錯誤數值多項式 $\omega(z)$ ，進行 Chien 搜尋。對於延伸成分以外的符號，則計算錯誤位置和錯誤數值，同時計數藉由此 Chien 搜尋所檢查出的錯誤個數 c ，而在其與所估算錯誤個數 e 不一致時，將旗標 F 設為"開"(ON)。另外，第 5 圖則表示此 Chien 搜尋處理程序的流程圖。

此處，在上述歐氏演算法計算步驟 ST13 中所計算出的錯誤位置多項式 $\sigma(z)$ 和錯誤數值多項式 $\omega(z)$ ，則是

(請先閱讀背面之注意事項再填寫本頁)

訂

五、發明說明 (24)

分別由下面的第(47)式和第(48)式所提供之。

$$\sigma(z) = \sigma_0 + \sigma_1 z + \sigma_2 z^2 + \sigma_3 z^3 \quad (47)$$

$$\omega(z) = \omega_0 + \omega_1 z + \omega_2 z^2 \quad (48)$$

在此 Chien 搜尋步驟 ST14 開始時，首先將用來計數此 Chien 搜尋中檢出之錯誤個數 c 的計數值，初始化為 0(步驟 ST140)。接著則是對錯誤個數估算步驟 ST12 中所估算出在延伸成分之外的錯誤個數 e，判斷其是否為 0(步驟 ST141)。接著將變數 i 初始化為 0 後(步驟 ST142)，利用下面的第(49)式和第(51)式，求出在此 Chien 搜尋的處理所需要的值 s、x 和 y(步驟 ST143)。

$$s = \sigma_0 + \sigma_1 + \sigma_2 + \sigma_3 \quad (49)$$

$$x = \omega_0 + \omega_1 + \omega_2 \quad (50)$$

$$y = \sigma_1 + \sigma_3 \quad (51)$$

接著判斷所得到的值 s 是否為 0(步驟 ST144)。若是判斷的結果是 s 為 0 時，則將此時變數 i 的值設定在錯誤位置 el[c]，同時將上述中值 x 除以值 y 的商設定在錯誤數值 ev[c](步驟 ST145)，並將錯誤個數 c 增加 1(步驟 ST146)。接著判斷變數 i 是否達到 n-2(步驟 ST147)。若沒有達到 n-2 時，則遞增變數 i，同時分別利用下面的第(52)式更新 $\sigma_0 \sim \sigma_3$ 以及下面的第(53)式更新 $\omega_0 \sim \omega_3$ (步驟 ST148)，回到步驟 ST143 進行上述的處理。

$$\sigma_j = \alpha^{-j} \sigma_j \quad (j=0,1,2,3) \quad (52)$$

$$\omega_j = \alpha^{-j-1} \omega_j \quad (j=0,1,2) \quad (53)$$

此處， α 為構成高氏場之原始方程式的根。

(請先閱請背面之注意事項再填寫本頁)

五、發明說明 (25)

另外在上述步驟 ST144 中，若判斷出 s 不為 0 時，則跳過步驟 ST145 和步驟 ST146 的處理，直接進行步驟 ST147 的判斷。

另外在上述步驟 ST147 中，若判斷出變數 i 達到 $n-2$ 時，則將此 Chien 搜尋所檢測出之錯誤個數 c 目前值，和在錯誤個數估算步驟 ST12 中所估算出的錯誤個數 e 值，進行比較(步驟 ST149)。若結果是兩者相等，則結束此時的 Chien 搜尋處理；若不相等時，則在將旗標 F 設為 "開"($F=1$)之後(步驟 ST150)，結束此 Chien 搜尋處理。

當 Chien 搜尋步驟 ST14 中所檢測出的錯誤個數 c 與先前在錯誤個數估算步驟 ST12 中所估算出的錯誤個數 e 一致時，則在錯誤訂正步驟 ST15 中會訂正錯誤並且輸出。而第 6 圖即表示此錯誤訂正處理程序的流程圖。處理一開始，首先判斷旗標 F 的開/關狀態(步驟 ST151)。此時若是先前在錯誤個數估算步驟 ST12 中所估算的錯誤個數 e ，與 Chien 搜尋步驟 ST14 中所檢測出的錯誤個數 c 一致時，則旗標為 "關"($F=0$)的狀態。

如果判斷結果表示旗標 F 不是 "開" 狀態($F=1$)，則在將變數 i 初始設定為 0(步驟 ST152)之後，進行變數 i 是否達到所檢測出之錯誤個數 c 的判斷(步驟 ST153)。如果判斷結果表示變數 i 並未達到錯誤個數 c ，則是對收訊字組 r_i ($i=0, 1, \dots, n-2$)之中，以基於此 i 值之錯誤位置 $e_{\text{t}}[i]$ 為附標(suffix)的符號 $r_{e_{\text{t}}[i]}$ ，依據下面的第(54)

(請先閱讀背面之注意事項再填寫本頁)

訂

五、發明說明 (26)

式將其值加上錯誤數值 $ev[i]$ 的方式，訂正其錯誤(步驟 ST154)。

$$r_{el[i]} = r_{el[i]} + ev[i] \quad (54)$$

之後，遞增變數 i (步驟 ST155)，回到步驟 ST153 重複上述的處理。當在步驟 ST153 中判斷出變數 i 達到所檢測出的錯誤個數 c 時，則輸出訂正錯誤後之收訊字組並且結束這一連串的處理。另外，當 Chien 搜尋步驟 ST14 中所檢測出的錯誤個數 c 與錯誤個數估算步驟 ST12 中所估算出的錯誤個數 e 不一致，亦即旗標 F 為“開”狀態($F=1$)時，因無法進行訂正而止於錯誤的檢測，並以此狀態結束其處理。

藉此，在第一實施例中，由於可以透過一次的歐氏演算法計算處理，來求得延伸 RS 碼之錯誤訂正解碼中所需要的錯誤位置多項式以及錯誤數值多項式，而且也能夠基於此錯誤位置多項式和錯誤數值多項式，利用一次的 Chien 搜尋處理來計算出錯誤位置和錯誤數值，所以可以達到高速進行延伸 RS 碼解碼操作的效果。

請先閱讀背面之注意事項再填寫本頁

訂

【第二實施例】

在上述的第一實施例中，雖然是針對延伸成分之外符號的錯誤數值計算加以說明，但是也能夠用來計算延伸成分符號的錯誤數值。第 7 圖即表示在本發明第二實施例具有如此功能之延伸 RS 碼錯誤訂正解碼方法中，Chien 搜尋步驟處理程序的詳細流程圖。

五、發明說明 (27)

接著說明其動作。

依照與第一實施例中相同的程序，首先在第 1 圖所示之徵候值產生步驟 ST11 中，計算收訊字組的徵候值，並在錯誤個數估算步驟 ST12 中估算延伸成分之外的錯誤個數。接著根據所估算的錯誤個數，變更初始值以及終止條件，並且在歐氏演算法計算步驟 ST13 中以歐氏演算法進行演算，計算出錯誤位置多項式以及錯誤數值多項式。

接著，根據此歐氏演算法計算步驟 ST13 中所計算出的錯誤位置多項式和錯誤數值多項式，在 Chien 搜尋步驟 ST14 中以 Chien 搜尋進行錯誤位置和錯誤數值的計算。如第 7 圖所示，在 Chien 搜尋處理開始時，先將錯誤個數的計數值初始化為 0(步驟 ST160)，再判斷所估算之延伸成分以外錯誤個數 e 是否為 0(步驟 ST161)。接著將變數 i 初始化為 0(步驟 ST162)，再利用第(49)式~第(51)式求得 s 、 x 、 y (步驟 ST163)，並判斷此 s 值是否為 0(步驟 ST164)。若 s 為 0，則將錯誤位置 $el[e]$ 設定成 i ，錯誤數值 $ev[c]$ 設定成 x/y (步驟 ST165)。而至此的程序則與第 5 圖所示，第一實施例中 Chien 搜尋處理的情況相同。

在此處若將 $\alpha^{-el[c]}$ (α 為構成高氏場之原始多項式的根)代入錯誤位置多項式 $\sigma(z)$ 的結果為 0 時，則利用下面的第(55)式，以此時的錯誤數值 $ev[c]$ 和錯誤位置 $el[c]$ 修正收訊字組的徵候值 S_5 。換言之，即是以此時徵候值 S_5 加

(請先閱讀背面之注意事項再填寫本頁)

訂

五、發明說明 (28)

以 $ev[c]\alpha^{6el[c]}$ 的方式，進行收訊字組徵候值 S_5 的修正(步驟 ST166)。

$$S_5 = S_5 + ev[c]\alpha^{6el[c]} \quad (55)$$

接著在將錯誤個數 c 增加 1 之後(步驟 ST167)，判斷變數 i 是否達到 $n-2$ (步驟 ST168)。若達到則遞增變數 i ，並在利用第(52)式和第(53)式更新 $\sigma_0 \sim \sigma_3$ 和 $\omega_0 \sim \omega_2$ 之後(步驟 ST169)，回到上述步驟 ST163 重複上述的處理。另外若 s 不為 0 時，則跳過上述步驟 ST165~步驟 ST167 的處理。當變數 i 的值達到 $n-2$ 時，則將上述錯誤個數 c 與所估算的錯誤個數 e 進行比較(步驟 ST170)，若兩者不相等，則將旗標 F 設為”開”狀態($F=1$)(步驟 ST171)。另外在步驟 ST167~步驟 ST171 的處理，亦與第 5 圖所示之第一實施例中 Chien 搜尋處理的情況相同。

在此處，對於延伸成分以外的所有收訊字組符號而言，當在徵候值 S_5 進行上述演算之結果為 0 以外的其他數值時，則延伸部分存在錯誤，其徵候值 S_5 的值即為延伸成分的錯誤數值。因此，在步驟 ST170 中檢測出兩者一致或是在步驟 ST171 中設定旗標 F 為”開”狀態之後，則進行徵候值 S_5 是否為 0 的判斷(步驟 ST172)。如果其結果是徵候值 S_5 不為 0，則將錯誤位置 $el[c]$ 設為 -1，錯誤數值 $ev[c]$ 則設為徵候值 S_5 (步驟 ST173)，再將錯誤個數 c 增加 1(步驟 ST174)，結束 Chien 搜尋的處理。而在步驟 ST161 中判斷出 $e=0$ 的情況下，或是在步驟 ST172 中判斷出 $S_5=0$ 的情況下，則以此狀態結束此 Chien 搜尋

(請先閱讀背面之注意事項再填寫本頁)

訂

五、發明說明 (29)

的處理。

其後之錯誤訂正步驟 ST15 中，則在進行延伸成分以外部分的錯誤訂正後，將對應於延伸成分的徵候值 S_5 加上修正後的值，結束錯誤訂正的操作。

藉此，在第二實施例中，由於能夠利用 Chien 搜尋中所計算出的錯誤位置和錯誤數值來修正收訊字組的徵候值，所以即使是在延伸成分內的符號，也可以達到輕易地計算出此錯誤數值的效果。

(一) 請先閱讀背面之注意事項再填寫本頁

【第三實施例】

在上述的實施例中，雖然是針對以錯誤個數估算步驟中所估算的錯誤個數而設定歐氏演算法計算動作中的初始值以及終止條件之情況來說明，但是也可以在錯誤個數估算步驟中判斷是否錯誤個數比延伸 RS 碼的最大錯誤訂正個數來得少或相等，再根據此判斷結果，變更歐氏演算法計算動作中的初始值以及終止條件。

第 8 圖表示在本發明第三實施例具有如此功能之延伸 RS 碼錯誤訂正解碼方法中處理程序的詳細流程圖。圖中，ST11 是徵候值產生步驟，用以對最小距離為 d 的延伸 RS 碼計算收訊字組的徵候值，等同於第 1 圖第一實施例中對應者。ST12 是錯誤個數估算步驟，用以根據所計算出的徵候值進行錯誤個數的估算時，判斷是否有錯誤，而在有錯誤時，判斷錯誤的個數是 $[(d-1)/2]$ 個或是低於 $[(d-1)/2]$ 個，其與第 1 圖第一實施例中相同標記

五、發明說明 (30)

者不同。ST16a 和 ST16b 則是根據此錯誤個數估算步驟 ST12 中所估算的結果，用以判斷其後所應實施處理的判斷步驟。

ST13a 和 ST13b 則是與第 1 圖所示之第一實施例中歐氏演算法計算步驟 ST13 對應的步驟。ST13a 是在判斷出錯誤個數估算步驟 ST12 所得錯誤個數 e 比 $[(d-1)/2]$ 個來得少之情況下，用來根據既定初始值和終止條件計算錯誤位置多項式和錯誤數值多項式之第一歐氏演算法計算步驟；ST13b 是在判斷出錯誤個數估算步驟 ST12 所得錯誤個數 e 等於 $[(d-1)/2]$ 個之情況下，用來根據既定初始值和終止條件，以不同於第一歐氏演算法計算步驟 ST13a 的方式，計算錯誤位置多項式和錯誤數值多項式之第二歐氏演算法計算步驟。

ST14 是 Chien 搜尋步驟，其利用第一歐氏演算法計算步驟 ST13a 和第二歐氏演算法計算步驟 ST13b 所得到的錯誤位置多項式以及錯誤數值多項式，計算錯誤位置和錯誤數值，等同於第 1 圖附以相同標記之步驟。ST17 則是根據此 Chien 搜尋步驟 ST14 所檢測出的錯誤個數，以及在第一歐氏演算法計算步驟 ST13a 和第二歐氏演算法計算步驟 ST13b 所得到錯誤位置多項式之階數，判斷是否可以進行錯誤訂正之判斷步驟。ST15a 和 ST15b 則是第 1 圖所示第一實施例中錯誤訂正步驟 ST15 的對應步驟。ST15a 的錯誤訂正操作步驟，是在判斷步驟 ST17 檢測出兩者一致的場合中，用來訂正錯誤的符號。ST15b

(請先閱讀背面之注意事項再填寫本頁)

訂

五、發明說明 (31)

的錯誤檢出操作步驟，則是在判斷步驟 ST17 檢測出兩者不一致的場合中，因無法進行訂正而僅檢測出錯誤的位置，並且輸出收訊字組的原來狀態。

接著說明其動作。

而此時也是針對最小距離為 7 的延伸 RS 碼，亦即檢查符號數量為 6 個符號者進行說明。

首先，在徵候值產生步驟 ST11 中，根據第一實施例中所說明的順序，計算收訊字組的徵候值 S_i 。接著在錯誤個數估算步驟 ST12 中，根據徵候值產生步驟 ST11 中所計算出的徵候值 S_i ，判斷錯誤個數為 0 個，或是非 0 但比 3 個少(2 個以下)，或是 3 個的任一種情況。第 9 圖即表示此錯誤個數估算步驟 ST12 之處理程序詳細流程圖。

當此錯誤個數估算步驟 ST12 之處理開始時，首先判斷在徵候值產生步驟 ST11 中所計算出的徵候值 $S_i(i=0, 1, \dots, 5)$ 是否全部為 0(步驟 ST180)。若全部為 0 則判斷沒有發生錯誤，設定錯誤個數 e 為 0(步驟 ST181)，並且結束此錯誤個數估算步驟 ST12 之處理。而在判斷出有錯誤時，則與第一實施例的情況相同，使用第(40)式~第(42)式，以收訊字組的徵候值 S_i 計算出 $T_1 \sim T_3$ (步驟 ST182)，同時對所得到的 $T_1 \sim T_3$ ，利用第(43)式計算出 U 值(步驟 ST183)。

接著進行所計算出之 U 值是否為 0 的判斷(步驟 ST184)。當 U 為 0 則估算錯誤個數比 3 來得少，而設定

(請先閱讀背面之注意事項再填寫本頁)

訂

五、發明說明 (32)

錯誤個數 e 為 2(步驟 ST185)，並且結束此錯誤個數估算步驟 ST12 的處理。另外 U 不為 0 則估算錯誤個數為 3 個，而設定錯誤個數 e 為 3(步驟 ST186)，並且結束此錯誤個數估算步驟 ST12 的處理。

藉由此錯誤個數估算步驟 ST12 之估算操作而設定之錯誤個數 e ，則分別在判斷步驟 ST16a 和判斷步驟 ST16b 與 0 和 2 進行比較。若比較的結果為 $e=0$ ，則因為推測沒有錯誤，所以以目前狀態結束處理。另外，若 $e=2$ 則因為推測錯誤個數比 3 個來得少，所以進入第一歐氏演算法計算步驟 ST13a，並且以下面所示之第(56)式和第(57)式做為用來計算錯誤位置多項式和錯誤數值多項式之初始值，以下面所示之第(58)式做為終止條件，利用歐氏演算法進行錯誤位置多項式和錯誤數值多項式。

$$R_{-1}(z) = z^5 \quad (56)$$

$$R_0(z) = S_4 z^4 + S_3 z^3 + S_2 z^2 + S_1 z + S_0 \quad (57)$$

$$\deg R_i(z) < 2 \quad (58)$$

若 $e=3$ 則因為推測錯誤個數為 3 個，所以進入第二歐氏演算法計算步驟 ST13b，並且以下面所示之第(59)式和第(60)式做為用來計算錯誤位置多項式和錯誤數值多項式之初始值，以下面所示之第(61)式做為終止條件，利用歐氏演算法進行錯誤位置多項式和錯誤數值多項式。

$$R_{-1}(z) = z^6 \quad (59)$$

(請先閱讀背面之注意事項再填寫本頁)

訂

五、發明說明 (33)

$$R_0(z) = S_5 z^5 + S_4 z^4 + S_3 z^3 + S_2 z^2 + S_1 z + S_0 \quad (60)$$

$$\deg R_i(z) < 3 \quad (61)$$

接著在 Chien 搜尋步驟 ST14 中，根據在此第一歐氏演算法計算步驟 ST13a 或是第二歐氏演算法計算步驟 ST13b 所計算出的錯誤位置多項式和錯誤數值多項式，與第一實施例的情況相同，對延伸成分以外之符號進行錯誤位置和錯誤數值的計算。接著在判斷步驟 ST17 中，比較上述計算出的錯誤位置多項式階數 $\deg\sigma(z)$ 以及利用 Chien 搜尋步驟 ST14 所檢測出的錯誤個數 c。若其一致時，則在錯誤訂正操作步驟 ST15a 中訂正收訊字組的錯誤並且輸出，結束此一處理。另一方法，當兩者不一致時，在錯誤檢出操作步驟 ST15b 中則無法訂正而止於錯誤的檢測，以收訊字組原來狀態輸出後，結束此一處理。

藉此，在第三實施例中，由於是根據在收訊字組之延伸成分以外部分所估算之錯誤個數，比延伸 RS 碼之最大錯誤訂正個數來得少或是相同的情況之判斷結果，來變更歐氏演算法計算操作中的初始值以及終止條件，而能夠計算出錯誤位置以及錯誤數值，所以可以達到減少收訊字組解碼時所需步驟的效果。

【第四實施例】

雖然在上述第三實施例之錯誤個數估算步驟中，是判斷少於或是等於延伸 RS 碼之最大錯誤訂正個數之任一者的情況來說明，但是也可以進行少於、等於或大於

(請先閱讀背面之注意事項再填寫本頁)

訂

五、發明說明 (34)

延伸 RS 碼之最大錯誤訂正個數的判斷，而在判斷大於的情況下，則不執行歐氏演算法計算操作以及 Chien 搜尋操作。

第 10 圖表示在本發明第四實施例具有如此功能之延伸 RS 碼錯誤訂正解碼方法中程序的詳細流程圖。圖中，ST12 是錯誤個數估算步驟，其根據微候值產生步驟 ST11 所計算出的微候值進行錯誤個數的估算時，判斷是否有錯誤，而在有錯誤時，判斷錯誤的個數是小於 $[(d-1)/2]$ 個、等於 $[(d-1)/2]$ 個或是大於 $[(d-1)/2]$ 個，其與第 8 圖第三實施例中相同標記者不同。ST16c 則與 ST16a 和 ST16b 相同，根據此錯誤個數估算步驟 ST12 中所估算的結果，用以判斷其後所應實施處理的判斷步驟，而在錯誤個數大於 $[(d-1)/2]$ 個時，則跳過歐氏演算法計算處理和 Chien 搜尋操作。另外其他步驟，則附上與第 8 圖所示第三實施例之對應步驟相同的標記，並且省略其說明。

接著說明其動作。

首先在微候值產生步驟 ST11 中，依照第一實施例所說明的順序，進行收訊字組的微候值 S_i 計算。接著在錯誤個數估算步驟 ST12 中，根據微候值產生步驟 ST11 中所計算出的微候值 S_i ，判斷錯誤個數為 0 個，或是非 0 但比 3 個少(2 個以下)，或是 3 個，或是比 3 個多(4 個以上)中的任一種情況。此處，第 11 圖即表示此錯誤個數估算步驟 ST12 之處理程序詳細流程圖。

當此錯誤個數估算步驟 ST12 之處理開始時，與第

(請先閱讀背面之注意事項再填寫本頁)

訂

五、發明說明 (35)

三實施例一樣，首先判斷在徵候值產生步驟 ST11 中所計算出的徵候值 S_i 是否全部為 0(步驟 ST190)。若全部為 0 則判斷沒有發生錯誤，設定錯誤個數 e 為 0(步驟 ST191)，並且結束此錯誤個數估算步驟 ST12。另一方面，在判斷出有錯誤時，則判斷是否下面第(62)式~第(64)式中之一成立(步驟 ST192)。

$$S_0 = S_1 = S_2 = 0 \quad (62)$$

$$S_1 = S_2 = S_3 = 0 \quad (63)$$

$$S_2 = S_3 = S_4 = 0 \quad (64)$$

當判斷結果顯示上述第(64)~(64)式中任一式成立時，則推定錯誤個數比 3 個多，而設定錯誤個數 e 為 4(步驟 ST193)，並且結束此錯誤個數估算步驟 ST12。

另一方面，如果上述第(62)式~第(64)式中沒有任何一式成立時，後面則與第三實施例的情況相同，亦即利用第(40)式~第(42)式計算 $T_1 \sim T_3$ (步驟 ST194)，利用第(43)式計算相對於此 $T_1 \sim T_3$ 的 U 值(步驟 ST195)，判斷是否所計算出的 U 值為 0(步驟 ST196)。接著，在 U 為 0 的情況下，則推定錯誤個數比 3 個少，並且設定錯誤個數 e 為 2(步驟 ST197)；而在 U 不為 0 的情況下，則推定錯誤個數為 3 個，並且設定錯誤個數 e 為 3(步驟 ST198)，隨後則結束此錯誤個數估算步驟 ST12 的處理。

另外，錯誤個數估算步驟 ST12 並非僅能依照上述第 11 圖所示之處理順序進行，舉例來說也可以採用以下所述之其他順序來處理。亦即，利用徵候值 S_i 全部不為

(請先閱讀背面之注意事項再填寫本頁)

訂

五、發明說明 (36)

0 等等方式來檢查錯誤的情況下，如果第(43)式所計算出的 U 值不為 0 時則估算錯誤的個數為 3 個。另外，如果上述的 U 值為 0，則視第(40)式~第(42)式所計算之 $T_1 \sim T_3$ 是否全部為 0；若是全部為 0 以外的情況則推定產生少於 3 個的錯誤，此以外的情況則推定產生多於 3 個的錯誤。

藉由此錯誤個數估算步驟 ST12 之估算操作所設定的錯誤個數 e 則分別在判斷步驟 ST16a、判斷步驟 ST16c、判斷步驟 ST16b 中與 0、4、2 進行比較。由於如果比較結果是 $e=0$ 時則推測沒有錯誤，所結束此一連串的處理。另外，由於若 $e=4$ 時是推測錯誤個數比 3 來得多，所以跳過歐氏演算法計算以及 Chien 搜尋的操作，在錯誤檢出操作步驟 ST15b 中則無法訂正而止於錯誤的檢測，以收訊字組原來狀態輸出後，結束此一處理。

另外此外的情況則與第三實施例的情況相同，由於 $e=2$ 時估計錯誤個數少於 3 個，所以在第一歐氏演算法計算步驟 ST13a 中，以第(56)式、第(57)式為初始值；並以第(58)式為終止條件，進行錯誤位置多項式以及錯誤數值多項式的計算。另外，由於 $e=3$ 時估計錯誤個數為 3 個，所以在第二歐氏演算法計算步驟 ST13b 中，以第(59)式、第(60)式為初始值，並以第(61)式為終止條件，進行錯誤位置多項式以及錯誤數值多項式的計算。

其後則與第三實施例一樣，在 Chien 搜尋步驟 ST14 中，根據這些錯誤位置多項式以及錯誤數值多項式，對

(請先閱讀背面之注意事項再填寫本頁)

註

五、發明說明 (37)

延伸成分以外的符號進行錯誤位置以及錯誤數值的計算。若在判斷步驟 ST17 中，研判出上述錯誤位置多項式的階數 $\deg\alpha(z)$ 和 Chien 搜尋步驟 ST14 所檢測出的錯誤個數一致的情況時，則在錯誤訂正操作步驟 ST15a 中訂正並且輸出收訊字組的錯誤。另外，若判定不一致的情況時，在錯誤檢出操作步驟 ST15b 中則無法訂正而止於錯誤的檢測，以收訊字組原來狀態輸出後，結束此一處理。

(請先閱讀背面之注意事項再填寫本頁)

訂

藉此，在第四實施例中，是在收訊字組之延伸成分以外部分所估算之錯誤個數時，比起延伸 RS 碼之最大錯誤訂正個數來得少、或是相等、或是大於之判斷結果，而在錯誤個數比最大錯誤訂正個數來得大時，則能夠跳過歐氏演算法計算操作以及 Chien 搜尋操作的處理，所以可以達到減少收訊字組解碼時所需步驟的效果。

【第五實施例】

在上述各實施例中雖是針對延伸 RS 碼的錯誤訂正解碼方法來做說明，但是也能夠據此來實現錯誤訂正解碼裝置。第 12 圖表示在本發明第五實施例具有如此功能之延伸 RS 碼錯誤訂正解碼裝置的方塊圖。圖中，1 為根據收訊字組產生微候值的微候值產生裝置，2 為根據在此微候值產生裝置 1 所計算出的微候值，用以估算收訊字組之延伸成分以外部分之錯誤個數的錯誤個數估算裝置。

五、發明說明 (38)

3 為基於錯誤個數估算裝置 2 中所估算出的錯誤個數來設定初始值和終止條件，並且藉由歐氏演算法來計算錯誤位置多項式以及錯誤數值多項式之歐氏演算法計算裝置。4 為對應於歐氏演算法計算裝置所計算出的錯誤位置多項式以及錯誤數值多項式，進行 Chien 搜尋，用以計算錯誤位置和錯誤數值的 Chien 搜尋裝置。5 為用來儲存所輸入之收訊字組的收訊字組記憶裝置。6 為用來儲存 Chien 搜尋裝置 4 中所計算之錯誤位置和錯誤數值的計算結果儲存裝置。7 為根據計算結果儲存裝置 6 中所儲存之錯誤位置和錯誤數值，用來訂正收訊字組儲存裝置 5 中所儲存之收訊字組錯誤的錯誤訂正裝置。

接著說明其操作。

輸入的收訊字組被送到徵候值產生裝置 1 和收訊字組儲存裝置 5。徵候值產生裝置 1 在接收收訊字組之後，則計算此收訊字組的徵候值 $S_i (i=0, 1, \dots, 5)$ ，並且送到錯誤個數估算裝置 2 以及歐氏演算法計算裝置 3。另外在收訊字組儲存裝置 5 中則儲存輸入之收訊字組的原來狀態。

在錯誤個數估算裝置 2 中，則根據徵候值產生裝置 1 所計算的徵候值 S_i 之中，不受延伸成分所影響的徵候值 S_0, S_1, \dots, S_4 ，估算延伸成分以外部分的錯誤個數。此錯誤個數估算裝置 2 所估算出的錯誤個數則被送到歐氏演算法計算裝置 3。在歐氏演算法計算裝置 3 中，則是基於此錯誤個數的估算值以及徵候值產生裝置 1 所計

(請先閱讀背面之注意事項再填寫本頁)

訂

五、發明說明 (39)

算的徵候值，設定用來計算錯誤位置多項式以及錯誤數值多項式之初始值和終止條件，並且藉由歐氏演算法進行錯誤位置多項式以及錯誤數值多項式的計算。

在 Chien 搜尋裝置 4 中，則是根據歐氏演算法計算裝置 3 所計算的錯誤位置多項式和錯誤數值多項式，進行 Chien 搜尋的操作，計算錯誤位置以及此位置上所產生的錯誤數值，並將其儲存在計算結果儲存裝置 6 中。接著錯誤訂正裝置 7 根據儲存在計算結果儲存裝置 6 中的錯誤位置以及錯誤數值之資訊，依序輸出儲存在收訊字組儲存裝置 6 中的收訊字組內容，並且將錯誤位置所對應的符號加上在此位置上所產生的錯誤數值，藉此訂正錯誤。

藉此，在第五實施例中，由於能夠以歐氏演算法計算裝置一次的處理而求得在延伸 RS 碼的錯誤訂正解碼所需要的錯誤位置多項式和錯誤數值多項式，並且也能夠以 Chien 搜尋裝置一次的處理而基於此錯誤位置多項式和錯誤數值多項式，計算出錯誤位置和錯誤數值，因此可以達到高速進行延伸 RS 碼之解碼操作的效果。

【第六實施例】

在上述之第六實施例中，雖然係以計算延伸成分以外之符號的錯誤數值情況加以說明，但是也可以應用於計算延伸成分的錯誤數值之情況。第 6 圖即為本發明第六實施例中具有如此功能之延伸 RS 碼的錯誤訂正解碼

(請先閱讀背面之注意事項再填寫本頁)

訂

五、發明說明 (40)

裝置，與第 12 圖中相同部分則加上相同的符號，並省略其說明。圖中，8 是徵候值修正裝置，其藉由高氏場上的乘法運算和加法運算而形成，並且根據 Chien 搜尋裝置 4 所計算的錯誤位置和錯誤數值，修正收訊字組的徵候值。

接著說明其動作。

首先收訊字組輸入到徵候值產生裝置 1 和收訊字組儲存裝置 5，在徵候值產生裝置 1 中進行收訊字組的徵候值 S_i 計算，並且在收訊字組儲存裝置中記憶此收訊字組。在錯誤個數估算裝置 2 中，根據在徵候值產生裝置 1 中所算出的徵候值內不影響延伸成分值的徵候值 S_0 、 S_1 、…、 S_4 ，估算延伸成分以外部分的錯誤個數，而在歐氏演算法計算裝置 3 中，則根據此錯誤個數的估算值設定初始值和終止條件，並且藉由歐氏演算法計算錯誤位置多項式和錯誤數值多項式。將此錯誤位置多項式和錯誤數值多項式送到 Chien 搜尋裝置 4 中，以計算錯誤位置以及此位置上的錯誤數值，儲存到計算結果儲存裝置 6 中。而至此的動作與第五實施例中的情況相同。

將 Chien 搜尋裝置 4 所計算出的錯誤位置和錯誤數值，送到徵候值修正裝置 8 中，徵候值修正裝置 8 是對徵候值產生裝置 1 中所計算而包含延伸成分之徵候值 S_5 ，根據其錯誤位置和錯誤數值進行高氏場的乘法加法運算，修正此徵候值 S_5 。對所有延伸成分以外符號計算其錯誤位置和錯誤數值的結果，則是在徵候值 S_5 所修正

(請先閱讀背面之注意事項再填寫本頁)

良
訂

五、發明說明 (41)

值不為 0 的情況下，以此延伸成分的位置做為錯誤位置，並以所修正的徵候值 S_5 做為其錯誤數值，將其儲存在計算結果儲存裝置 6 中。

錯誤修正裝置 7 則與第五實施例的情況相同，根據此計算結果儲存裝置 6 中所儲存的錯誤位置和錯誤數值之資訊，依序輸出儲存在收訊字組儲存裝置 5 中所儲存的收訊字組內容，而對存在錯誤的符號值加上在此位置上所產生的錯誤數值，訂正其錯誤。

藉此，在第六實施例中，由於能夠利用徵候值修正裝置中的高氏場之乘法加法運算，並且依據 Chien 搜尋中所計算出的錯誤位置和錯誤數值，修正收訊字組的徵候值，所以也能夠輕易地達到計算出延伸成分錯誤數值的效果。

(請先閱讀背面之注意事項再填寫本頁)

訂

【第七實施例】

在上述第五和第六實施例中所說明的情況，雖然是以錯誤個數估算裝置所估算出的錯誤個數，來設定歐氏演算法計算動作中的初始值和終止條件，但是在錯誤個數估算裝置中，也可以先判斷比延伸 RS 碼的最大錯誤訂正個數來得少或相同，再根據此判斷結果，變更歐氏演算法計算動作中的初始值和終止條件。

第 14 圖即為本發明第七實施例中具有如此功能之延伸 RS 碼的錯誤訂正解碼裝置之方塊圖。圖中，9 是根據徵候值產生裝置 1 所計算出的徵候值，估算收訊字組

五、發明說明 (42)

中延伸成分以外部分的錯誤個數之錯誤個數估算裝置，另外其與第 12 圖附上標記 2 的部分不同的是，其藉由高氏場乘法裝置、加法裝置以及儲存裝置所形成，不僅用來估算錯誤個數，同時在收訊字組中延伸成分以外部分存在錯誤時，會判斷此錯誤個數是比 $[(d-1)/2]$ 來得少或是相同的情況。3 則是對應於錯誤個數估算裝置 9 的判斷結果來設定初始值和終止條件，並且據此進行錯誤位置多項式和錯誤數值多項式之計算的歐氏演算法計算裝置。另外，其他部分則與第五實施例所示者相同，附上與第 12 圖相同的標記，並且省略其說明。

接著說明其操作。

與第五實施例的情況一樣，首先收訊字組輸入到微候值產生裝置 1 和收訊字組儲存裝置 5，在微候值產生裝置 1 中進行收訊字組微候值 S_i 的計算，同時在收訊字組儲存裝置 5 中儲存原本狀態的收訊字組。

在錯誤個數估算裝置 9 中，根據在微候值產生裝置 1 中所產生的微候值 S_i 中不受延伸成分所影響的微候值 S_0, S_1, \dots, S_4 ，判斷延伸成分以外的錯誤個數是否為 0 個，而在不是 0 個的情況下，則判斷是少於 3 個(2 個以下)或是 3 個的那一種情況。另外，此判斷則可以採用像是第三實施例說明中第 9 圖所示流程圖的演算法。在判斷結果顯示錯誤個數不為 0 的情況下(存在錯誤的情況)，則在歐氏演算法計算裝置 3 中，根據在錯誤個數估算裝置 9 所判斷的錯誤個數之條件，設定用來計算錯誤位置

(請先閱讀背面之注意事項再填寫本頁)

訂

五、發明說明 (43)

多項式和錯誤數值多項式之初始值以及終止條件，並且藉由歐氏演算法，進行錯誤位置多項式和錯誤數值多項式的計算。

所算出的錯誤位置多項式和錯誤數值多項式則被送到 Chien 搜尋裝置 4 中，其後則與第五實施例相同，在 Chien 搜尋裝置 4 中利用此錯誤位置多項式和錯誤數值多項式，計算出錯誤位置和此位置上所產生的錯誤數值，並儲存於計算結果儲存裝置 6 中。再根據儲存於計算結果儲存裝置 6 的錯誤位置和錯誤數值資訊，依序輸出收訊字組儲存裝置 5 中內容，而在錯誤訂正裝置 7 中，並將存在錯誤的符號加上此位置所產生的錯誤數值，訂正其錯誤。

藉此，在第七實施例中，由於是根據在錯誤個數估算裝置中所執行、比延伸 RS 碼之最大錯誤訂正個數來得少或是相同的情況之判斷結果，來變更歐氏演算法計算操作中的初始值以及終止條件，而計算出錯誤位置以及錯誤數值，所以可以達到減少收訊字組解碼時所需步驟的效果。

【第八實施例】

在上述第七實施例之錯誤個數估算裝置中，雖然是在存在錯誤的情況下，對於比延伸 RS 碼之最大錯誤訂正個數來得少或是等於中任一種情況之判斷來做說明，但是也可以判斷是少於、等於或大於延伸 RS 碼之最大

(請先閱讀背面之注意事項再填寫本頁)

訂

五、發明說明 (44)

錯誤訂正個數，而在判斷出大於的情況下，則不進行歐氏演算法計算裝置和 Chien 搜尋裝置中的處理操作。

第 15 圖即為本發明第八實施例中具有如此功能之延伸 RS 碼的錯誤訂正解碼裝置之方塊圖。對應部分則附上與第 14 圖相同的標記，並省略其說明。圖中，9 與第七實施例之第 14 圖中附上相同標記者不同，其在收訊字組之延伸成分以外部分存在錯誤時，判斷其錯誤個數是少於 $[(d-1)/2]$ 個，等於 $[(d-1)/2]$ 個和大於 $[(d-1)/2]$ 個中的何種情況。

以下說明其動作。

與第六實施例的情況一樣，首先收訊字組輸入到徵候值產生裝置 1 和收訊字組儲存裝置 5，在徵候值產生裝置 1 中進行收訊字組徵候值 S_i 的計算，同時在收訊字組儲存裝置 5 中儲存此收訊字組。

在錯誤個數估算裝置 9 中，根據在徵候值產生裝置 1 中所產生的徵候值 S_i 中不受延伸成分所影響的徵候值 S_0, S_1, \dots, S_4 ，判斷延伸成分以外的錯誤個數是否為 0 個，而在不是 0 個的情況下，則判斷是少於 3 個(2 個以下)、恰為 3 個、或是比 3 個多(4 個以上)的那一種情況。另外，此判斷則可以採用像是第四實施例說明中第 11 圖所示流程圖的演算法。在判斷結果顯示錯誤個數比 3 個多的情況下，則停止歐氏演算法計算裝置 3 中藉由歐氏演算法的錯誤位置多項式和錯誤數值多項式之計算處理，以及 Chien 搜尋裝置中藉由 Chien 搜尋的錯誤位置

(請先閱讀背面之注意事項再填寫本頁)

訂

五、發明說明 (45)

和錯誤數值之計算處理，並且將收訊字組儲存裝置 5 所儲存的收訊字組，藉由錯誤訂正裝置 7 輸出其原來的狀態。

另外，當判斷結果顯示錯誤個數比 3 個少，但是不為 0 時，則與第七實施例的情況一樣，根據在錯誤個數估算裝置 9 所判斷的錯誤個數之條件，在歐氏演算法計算裝置 3 中設定歐氏演算法的初始值以及終止條件，並且據此來進行錯誤位置多項式和錯誤數值多項式的計算。其後與第七實施例的情況一樣，將此錯誤位置多項式和錯誤數值多項式送到 Chien 搜尋裝置 4 中，計算出錯誤位置和此位置上所產生的錯誤數值，並儲存於計算結果儲存裝置 6 中。再根據儲存於計算結果儲存裝置 6 的錯誤位置和錯誤數值資訊，依序輸出收訊字組儲存裝置 5 中內容，而在錯誤訂正裝置 7 中，並將存在錯誤的符號加上此位置所產生的錯誤數值，訂正其錯誤。

藉此，在第八實施例中，由於是根據在錯誤個數估算裝置中所執行、比延伸 RS 碼之最大錯誤訂正個數來得少、相等或是來得多的情況之判斷結果，來變更歐氏演算法計算操作中的初始值以及終止條件，而計算出錯誤位置以及錯誤數值，所以可以達到減少收訊字組解碼時所需步驟的效果。

【第九實施例】

本發明的第九實施例係針對一階延伸 RS 碼之錯誤

(請先閱讀背面之注意事項再填寫本頁)

訂

閱

五、發明說明 (46)

訂正方法加以說明。此處則是先從一階延伸 RS 碼之錯誤訂正原理開始說明。

在此第九實施例中，係採用在習知技術中所說明之碼長 n 、資訊符號數 k 、最小距離 $d=n-k+1=2t+1$ (t 為正整數) 的一階延伸 RS 碼來進行。另外為了簡化起見，高氏場的幕數設為 2。

此處，做為一階延伸 RS 碼字組的傳送字組 $C=(c_{n-2}, c_{n-3}, \dots, c_0, c_{-1})$ ，會在傳送路徑中加上錯誤字組 $E=(e_{n-2}, e_{n-3}, \dots, e_0, e_{-1})$ 。假設產生 $t=[(d-1)/2]$ 個符號以下的錯誤，則上述錯誤字組 E 中非零部分者則在 t 個符號以下。但是在延伸成分所產生的錯誤則不算成一個符號，而是算成 $1/2$ 個符號。此時，收訊字組 $R=(r_{n-2}, r_{n-3}, \dots, r_0, r_{-1})$ 可藉由下面第(65)式加以計算。

$$R = C + E = (c_{n-2} + e_{n-2}, c_{n-3} + e_{n-3}, \dots, c_0 + e_0, c_{-1} + e_{-1}) \quad (65)$$

為了由此收訊字組 R 估算出所傳送的傳送字組 C ，所以利用下面所示之第(66)式，計算包含錯誤資訊之徵候值 $S=(S_0, S_1, \dots, S_{2t-2}, S_{2t-1})$ 。而 H 即為第(5)式之配類陣列。

$$S = H^T R = H(C + E) = H^T E \quad (66)$$

換言之，徵候值可以利用下面所示之第(67)式和第(68)式而得。

$$S_j = e_{n-2} \alpha^{(b+j)(n-2)} + e_{n-3} \alpha^{(b+j)(n-3)} + \dots + e_1 \alpha^{b+j} + e_0 \quad (j = 0, 1, 2, \dots, 2t-2) \quad (67)$$

$$S_{2t-1} = e_{n-2} \alpha^{(b+2t-1)(n-2)} + e_{n-3} \alpha^{(b+2t-1)(n-3)} + \dots + e_1 \alpha^{b+2t-1} + e_0 + e_{-t} \quad (68)$$

(請先閱讀背面之注意事項再填寫本頁)

訂

閱

五、發明說明 (47)

此處若延伸成分上產生了錯誤($e_i \neq 0$)，則延伸成分上錯誤所影響的徵候值即為 S_{2t-1} 。現在內部碼中所產生的錯誤個數則設為 $s \leq t$ (而在延伸成分內產生錯誤的情況時，則 $s \leq t-1$)，其錯誤位置的集合則是 $L = \{i | e_i \neq 0, 0 \leq i \leq n-2\}$ 。另外，此錯誤位置集合 L 的元素則是 $l[k](k=0, 1, \dots, s-1)$ 。對應的錯誤數值則為 $e_{l[k]}$ 。此時，上述的第(67)式和第(68)式則變為下面所示的第(69)式和第(70)式。

$$S_j = e_{l[0]} \alpha^{(b+j)/[0]} + e_{l[1]} \alpha^{(b+j)/[1]} + \dots + e_{l[s-1]} \alpha^{(b+j)/[s-1]} \quad (j = 0, 1, 2, \dots, 2t-2) \quad (69)$$

$$S_{2t-1} = e_{-1} + e_{l[0]} \alpha^{(b+2t-1)/[0]} + e_{l[1]} \alpha^{(b+2t-1)/[1]} + \dots + e_{l[s-1]} \alpha^{(b+2t-1)/[s-1]} \quad (70)$$

(請先閱讀背面之注意事項再填寫本頁)

徵候值多項式則以下面所示之第(71)式所構成，並將延伸成分所影響的 S_{2t-1} 設為常數項。

$$S(x) = S_{2t-1} + S_{2t-2}x + \dots + S_1x^{2t-2} + S_0x^{2t-1} \quad (71)$$

利用上述第(69)式和第(70)式，能夠調整此第(71)式成為下面所示之第(72)式。

$$\begin{aligned} S(x) &= e_{-1} + \sum_{j=0}^{2t-1} \sum_{k=0}^{s-1} e_{l[k]} \alpha^{(b+2t-1-j)/[k]} x^j \\ &= e_{-1} + \sum_{j=0}^{\infty} \sum_{k=0}^{s-1} e_{l[k]} \alpha^{(b+2t-1-j)/[k]} x^j \quad \text{mod } x^{2t} \\ &= e_{-1} + \sum_{k=0}^{s-1} e_{l[k]} \alpha^{(b+2t-1)/[k]} \sum_{j=0}^{\infty} \alpha^{-j/[k]} x^j \quad \text{mod } x^{2t} \\ &= e_{-1} + \sum_{k=0}^{s-1} e_{l[k]} \alpha^{(b+2t-1)/[k]} (1 + \alpha^{-1/[k]} x)^{-1} \quad \text{mod } x^{2t} \end{aligned} \quad (72)$$

此處，錯誤位置多項式以下面第(73)式的方式構成。

$$\sigma(x) = (1 + \alpha^{-l[0]} x)(1 + \alpha^{-l[1]} x) \cdots (1 + \alpha^{-l[s-1]} x) \quad (73)$$

將此第(73)式和上述的第(74)式兩邊相乘，即得到

訂

五、發明說明 (48)

下面所示的第(74)式。

$$S(x)\sigma(x) = e_{-1}\sigma(x) + \sum_{k=0}^{s-1} e_{l[k]} \alpha^{(b+2t-1)[k]} \prod_{j \neq k} (1 + \alpha^{-l[k]} x) \mod x^{2t} \quad (74)$$

此處將其右邊表示為 $\omega(x)$ ，上述第(74)式則變為下面的第(75)式，得到所謂的基本方程式。

$$\sigma(x)S(x) \equiv \omega(x) \mod x^{2t} \quad (75)$$

而此 $\omega(x)$ 雖然稱之為錯誤數值多項式，但是當延伸成分上產生錯誤時 ($e_{-1} \neq 0$)，第一項 $e_{-1}\sigma(x)$ 出現點則與一般的情況不同。此時，錯誤位置多項式 $\sigma(x)$ 的次數和錯誤數值多項式 $\omega(x)$ 的次數是一致的。雖然錯誤位置多項 $\sigma(x)$ 和錯誤數值多項式 $\omega(x)$ 都滿足上述的基本方程式，而當抽出其之外的特徵，假設 t 個符號以下的錯誤，則得到下面相關於次數的條件 1。而 \deg 則表示多項式的次數。

[條件 1]：當產生 $s(\leq t)$ 個符號錯誤時， $\deg \sigma = [s]$ 且 $\deg \omega \leq [s-1/2]$ (特別是雖然 $\deg \sigma \geq \deg \omega$ ，但是當延伸成分中產生錯誤時， $\deg \sigma = \deg \omega$)

另外，藉由上述的第(73)式和第(74)式，可得到下面的條件 2。

[條件 2]： $\sigma(x)$ 和 $\omega(x)$ 彼此互斥 (不具有共通根)。

另外，採用適當的多項式 $\phi(x)$ ，第(75)式所示之基本方程式等效於下面第(76)式中所表示者。

$$\phi(x)x^{2t} + \sigma(x)S(x) = \omega(x) \quad (76)$$

接著，以徵候值多項式為已知，可求取滿足上述基

(請先閱讀背面之注意事項再填寫本頁)

校
訂

五、發明說明 (49)

本方程式之錯誤位置多項式和錯誤數值多項式，並且對於 t 個符號錯誤的情況，證明其解的存在性和唯一性。

以下即以歐氏演算法表現此結構上的證明。

以下面的第(77)式和第(78)式來定義多項式組合 $(A_k(x), B_k(x), R_k(x))$, $k \geq -1$ 。

$$A_{-1}(x) = 1, \quad B_{-1}(x) = 0, \quad R_{-1}(x) = x^{2t} \quad (77)$$

$$A_0(x) = 0, \quad B_0(x) = 1, \quad R_0(x) = S(x) \quad (78)$$

而對於 $k \geq 1$ 者，則是依序地以 $R_{k-1}(x)$ 除以 $R_{k-2}(x)$ 時的商多項式為 $Q_k(x)$ ，再藉由下面的第(79)式~第(81)式來定義。

$$A_k(x) = A_{k-1}(x) + Q_k(x)A_{k-1}(x) \quad (79)$$

$$B_k(x) = B_{k-1}(x) + Q_k(x)B_{k-1}(x) \quad (80)$$

$$R_k(x) = R_{k-1}(x) + Q_k(x)R_{k-1}(x) \quad (81)$$

此時根據定義， $B_k(x)$ 的次數呈單調增加，即 $\deg B_k < \deg B_{k-1}$ ； $R_k(x)$ 的次數呈單調下降，即 $\deg R_{k-1} > \deg R_k$ 。

另外，藉由歸納法，可以利用下面所示之第(82)式~第(84)式，表示多項式組合 $(A_k(x), B_k(x), R_k(x))$ 的關係。

$$A_k(x)x^{2t} + B_k(x)S(x) = R_k(x) \quad (k \geq -1) \quad (82)$$

$$B_k(x)R_{k-1}(x) + B_{k-1}(x)R_k(x) = x^{2t} \quad (k \geq -1) \quad (83)$$

$$A_k(x)B_{k-1}(x) + A_{k-1}(x)B_k(x) = 1 \quad (k \geq -1) \quad (84)$$

此處以上述第(82)式為例說明。此第(82)式在 $k=-1$ ，0 時，根據第(77)式和第(78)式，明顯地會成立。另外，假設一直到 $k-1$ 都成立，而當 k 時也可藉由下述的第(85)式得知其成立。而在此公式推演中則採用了第(79)式~第

(請先閱讀背面之注意事項再填寫本頁)

訂

五、發明說明 (50)

(81)。

$$\begin{aligned}
 & A_k(x)x^{2t} + B_k(x)S(x) \\
 & = (A_{k-2}(x) + Q_k(x)A_{k-1}(x))x^{2t} + (B_{k-2}(x) + Q_k(x)B_{k-1}(x))S(x) \\
 & = A_{k-2}(x)x^{2t} + B_{k-2}(x)S(x) + Q_k(x)(A_{k-1}(x)x^{2t} + B_{k-1}(x)S(x)) \\
 & = R_{k-2}(x) + Q_k(x)R_{k-1}(x) \\
 & = R_k(x)
 \end{aligned} \tag{85}$$

同樣地，利用歸納法也可以表示上述第(83)式和第(84)式。另外，雖然藉由第(82)式可知，($B_k(x)$, $R_k(x)$)可以做為錯誤位置多項式和錯誤數值多項式的一個候選解，但是以下所示者，則可針對適當的 k 值提供真正的錯誤位置多項式和錯誤數值多項式。

首先注意 $B_k(x)$ 的次數呈單調增加， $R_k(x)$ 的次數呈單調下降，並且根據上述之第(83)式，下面所示之第(86)式會成立。

$$2t = \deg x^{2t} = \deg(B_k R_{k-1} + B_{k-1} R_k) = \deg B_k R_{k-1} = \deg B_k + \deg R_{k-1} \tag{86}$$

此處，由於 $\deg R_k$ 呈單調下降，所以存在一 k 值，滿足下述之第(87)式條件。

$$\deg R_{k-1} \geq t \text{ 且 } \deg R_k < t \tag{87}$$

此時藉由第(86)式，發現相關於 B_k 次數之下面第(88)式會成立。

$$\deg B_k = 2t - \deg R_{k-1} \leq t \tag{88}$$

第(82)式對此 k 值成立。而在第(76)式乘上 $B_k(x)$ ，並在第(82)式乘上 $\sigma(x)$ ，兩邊相加，即得到下面之第(89)式。

$$(B_k(x)\phi(x) + \sigma(x)A_k(x))x^{2t} = B_k(x)\omega(x) + \sigma(x)R_k(x) \tag{89}$$

(請先閱讀背面之注意事項再填寫本頁)

訂

五、發明說明 (51)

而在此第(89)式中，根據上述條件 1 以及第(87)式和第(88)式，此第(89)式的右邊是 $2t-1$ 次以下的多項式，而如果()內不是零時，則其左邊是 $2t$ 次以上的多項式，所以可以推論出 $()=0$ 。因此可得下面的第(90)式和第(91)式。

$$B_k(x)\phi(x) = \sigma(x)A_k(x) \quad (90)$$

$$B_k(x)\omega(x) = \sigma(x)R_k(x) \quad (91)$$

可是藉由上述的條件 2，由於 $\sigma(x)$ 和 $\omega(x)$ 彼此互斥（不具有共通根），所以存在適當的多項式 $\lambda(x)$ ，而得到下面所示的第(92)式~第(94)式。

$$B_k(x) = \lambda(x)\sigma(x) \quad (92)$$

$$R_k(x) = \lambda(x)\omega(x) \quad (93)$$

$$A_k(x) = \lambda(x)\phi(x) \quad (94)$$

另外，因為根據第(84)式， $A_k(x)$ 和 $B_k(x)$ 為彼此互斥，可知 $\lambda(x)$ 為一常數。因此，可以推論出 $(B_k(x), R_k(x))$ 除了常數倍的差異之外，可做為所需要的解。換言之， $(B_k(x)/B_k(0), R_k(x)/R_k(0))$ 即為真正的錯誤位置多項式 $\sigma(x)$ 和錯誤數值多項式 $\omega(x)$ 。

錯誤位置的集合 L 可藉由檢查 $B_k(x)$ 的根而求得。當求得錯誤位置集合 L 後，便能夠以下面所示之第(95)式，求得各 $l[k]$ 中錯誤的大小 $e_{l[k]}$ 。而式中的 σ' 為 σ 的形式微分。

$$e_{l[k]} = \frac{\omega(\alpha^{l[k]})}{\sigma'(\alpha^{l[k]})\alpha^{(b+2l)[k]}} \quad (95)$$

(請先閱讀背面之注意事項再填寫本頁)

訂

五、發明說明 (52)

但是當錯誤位置多項式 $\sigma(x)$ 的次數和錯誤數值多項式 $\omega(x)$ 的次數一致時，可知在延伸成分中存在錯誤。此錯誤的大小則可以藉由將錯誤位置多項式 $\sigma(x)$ 中最高次項的係數除以錯誤數值多項式 $\omega(x)$ 最高次項的係數而求得。

此處，第 16 圖表示第九實施例中一階延伸 RS 碼之錯誤訂正方法處理程序的流程圖，上述錯誤訂正方法所適用的情況，是針對 $b=1$ ，最小距離為 7 之高氏場 $GF(2^7)$ 上 $(128, 122)$ 一階延伸 RS 碼。圖中，ST21 是以收訊字組計算微候值之微候值產生步驟，ST22 為微候值多項式產生步驟，係以微候值產生步驟 ST21 所計算出的微候值中包含延伸成分的錯誤資訊者置於常數項的方式，產生微候值多項式。

ST23 為歐氏演算法計算步驟，係以微候值多項式產生步驟 ST22 所產生之微候值多項式，做為初始值，再藉由歐氏演算法計算錯誤位置多項式和錯誤數值多項式。ST24 則是用來比較歐氏演算法計算步驟中所得到之錯誤位置多項式和錯誤數值多項式次數的比較步驟。ST25 為 Chien 搜尋步驟，在比較步驟 ST24 中判斷出錯誤數值多項式的次數比錯誤位置多項式來得小時，對這些錯誤位置多項式和錯誤數值多項式進行 Chien 搜尋，以計算出錯誤位置和錯誤數值。ST26 為錯誤訂正步驟，其根據在 Chien 搜尋步驟 ST25 中所得到的錯誤位置和錯誤數值，訂正在收訊字組中的錯誤。

(請先閱讀背面之注意事項再填寫本頁)

衣

訂

五、發明說明 (53)

接著說明其動作。

首先在徵候值產生步驟 ST21 中，藉由第(6)式，以收訊字組計算徵候值 $S_i (i=0, 1, \dots, 5)$ 。接著在徵候值多項式產生步驟 ST22 中，將屬於延伸成分的徵候值 S_5 設為常數項，以下面第(96)的方式設定徵候值多項式。

$$S(x) = S_5 + S_4x + \dots + S_1x^4 + S_0x^5 \quad (96)$$

接著在歐氏演算法計算步驟 ST23 中，利用此徵候值多項式 $S(x)$ ，進行歐氏演算法的計算。第 17 圖表示此歐氏演算法計算步驟 ST23 之詳細處理程序的流程圖。如第 17 圖所示，首先利用下面的第(97)式和第(98)式，設定歐氏演算法的初始值(步驟 ST210)。

$$B_{-1}(x) = 0, \quad B_0(x) = 1 \quad (97)$$

$$R_{-1}(x) = x^6, \quad R_0(x) = S(x) \quad (98)$$

接著在將變數 i 設定為初始值 1 之後(步驟 ST211)，將 $R_{i-1}(x)$ 除以 $R_{i-2}(x)$ 時的商多項式做為 $Q(x)$ (步驟 ST212)，在下面的第(99)式和第(100)式中計算出 $R_i(x)$ 和 $B_i(x)$ (步驟 ST213)。

$$R_i(x) = Q_i(x)R_{i-1}(x) + R_{i-2}(x) \quad (99)$$

$$B_i(x) = Q_i(x)B_{i-1}(x) + B_{i-2}(x) \quad (100)$$

接著判斷藉此所計算出的 $R_i(x)$ 次數是否未滿 3(步驟 ST214)。若不是未滿 3 時，則遞增變數 i (步驟 ST215)，返回重複步驟 ST212 以下的處理。在步驟 ST214 中，判斷出 $R_i(x)$ 的次數未滿 3 時，則將 $\sigma(x) = B_i(x)/B_i(0)$ ， $\omega(x) = R_i(x)/R_i(0)$ (步驟 ST216)。並且結束此歐氏演算法。

(請先閱讀背面之注意事項再填寫本頁)

訂

五、發明說明 (54)

計算步驟 ST23。

接著在比較步驟 ST24 中，比較錯誤位置多項式 $\sigma(x)$ 的次數和錯誤數值多項式 $\omega(x)$ 的次數。其結果若顯示 $\deg\sigma(x) \geq \deg\omega(x)$ 不成立時，則做為錯誤的檢測，並輸出原來的收訊字組。另一方面，若 $\deg\sigma(x) \geq \deg\omega(x)$ 成立時，則在 Chien 搜尋步驟 ST25 中，對錯誤位置多項式 $\sigma(x)$ 和錯誤數值多項式 $\omega(x)$ 進行 Chien 搜尋，以計算錯誤位置和錯誤的大小。而在等號成立的情況下，則推測在延伸成分中存在有錯誤。

當在 Chien 搜尋步驟 ST25 中所計算出的錯誤個數與錯誤位置多項式之次數一致時，則在錯誤訂正步驟 ST26 中訂正此錯誤，並且輸出訂正後的結果。另外，當檢查出的錯誤個數與錯誤位置多項式的次數不一致時，則僅止於錯誤的檢測，並且輸出原來的收訊字組。

藉此，在習知使用歐氏演算法的一階延伸 RS 碼錯誤訂正方法中，最差的情況下需要進行兩次的歐氏演算法計算，所以會有解碼延遲的問題，但在第九實施例之一階延伸 RS 碼的錯誤訂正方法中，只需要進行一次的歐氏演算法計算，另外，利用比較歐氏演算法計算中所檢測出的錯誤位置多項式和錯誤數值多項式之次數，也很容易判斷出是否在延伸成分中產生了錯誤。

【第十實施例】

在上述第九實施例中，雖然是針對以歐氏演算法來

(請先閱讀背面之注意事項再填寫本頁)

訂

五、發明說明 (55)

進行一階延伸 RS 碼錯誤訂正的一階延伸 RS 碼錯誤訂正方法進行說明，但是也可以適用於用來進行一階延伸 RS 碼消失/錯誤訂正的一階延伸 RS 碼錯誤訂正方法。以下即對於本發明第十實施例，一階延伸 RS 碼之消失/錯誤訂正方法的原理加以說明。而在第十實施例的說明中，是採用習知技術中說明的碼長 n 、資訊符號數 k 、最小距離 $d=n-k+1=2t+1$ (t 為正整數)之一階延伸 RS 碼。另外為簡化起見，高氏場的幕數設為 2。

以輸入的收訊字組，計算出徵候值 $S_0, S_1, \dots, S_{2t-1}$ ，並藉此構成與第(71)式相同的徵候值多項式。另外，將利用收訊字組所伴隨的消失旗標而判斷消失的內部碼符號集合設為 M ，此集合 M 的元素則為 $m[k](k=0, 1, \dots, h-1)$ 。再將消失之外的隨機錯誤集合設為 L ，此集合 L 的元素則為 $l[k](k=0, 1, \dots, s-1)$ 。

現在假設 $2s+h \leq 2t$ 的條件成立，說明用來訂正消失和隨機錯誤的方法。但是在延伸成分中所產生的隨機錯誤不是算為 1 個，而是 $1/2$ 個。以消失位置的集合 M 中各元素 $m[k]$ ，構成消失位置係數 $\alpha^{-m[k]}$ ，利用此消失位置係數構成下面第(101)式所示的多項式，其稱為消失位置多項式。

$$\Lambda(x) = (1 + \alpha^{-m[0]}x)(1 + \alpha^{-m[1]}x) \cdots (1 + \alpha^{-m[h-1]}x) \quad (101)$$

另外，與第(73)式一樣，以延伸成分以外的隨機錯誤，構成錯誤位置多項式 $\sigma(x)$ 。此處與第九實施例的第(72)式相同，計算 $\sigma(x)\Lambda(x)S(x) \bmod x^{2t}$ ，可得到 $[h+s-1/2]$ 次

(請先閱讀背面之注意事項再填寫本頁)

討

五、發明說明 (56)

數以下的多項式。此多項式設為 $\pi(x)$ ，可得到與第九實施例相同，以下面第(102)式所示之基本方程式。

$$\sigma(x)\Lambda(x)S(x) = \pi(x) \mod x^{2t} \quad (102)$$

而 $\pi(x)$ 是包含在錯誤位置和消失位置中錯誤大小相關資訊的多項式。此多項式在以下則稱之為錯誤消失數值多項式。此處利用下面所示之第(103)式定義一多項式，稱之為修正徵候值多項式。

$$T(x) = \Lambda(x)S(x) \mod x^{2t} \quad (103)$$

根據定義，此多項式是在 $2t-1$ 次數以下的多項式。上述第(102)式的基本方程式可表示為下面的第(104)式。

$$\sigma(x)T(x) = \pi(x) \mod x^{2t} \quad (104)$$

另外，利用適當的多項式 $\varphi(x)$ ，可將此基本方程式等效表示為下面的第(105)式。

$$\varphi(x)x^{2t} + \sigma(x)T(x) = \pi(x) \quad (105)$$

另外，錯誤位置多項式 $\sigma(x)$ 和錯誤消失數值多項式 $\pi(x)$ 滿足上述的基本方程式，而當抽出其他的特徵後，可以得到相關於次數、下面所示之條件 3。而 \deg 表示多項式的次數。

[條件 3]： $\deg\sigma=[s]$ 時， $\deg\pi \leq [h+s-1/2]$ 。

另外與第九實施例之條件 2 一樣，可得到下面的條件 4。

[條件 4]： $\sigma(x)$ 和 $\pi(x)$ 彼此互斥(不具有共通根)。

滿足上述條件 3 和條件 4 的基本方程式之解，與第九實施例的情況一樣，可以藉由歐氏演算法解出。

(請先閱讀背面之注意事項再填寫本頁)

訂

五、發明說明 (57)

以下概略加以說明。

首先，歐氏演算法的初始值是以下面第(106)式和第(107)式的方式加以設定。

$$A_{-1}(x) = 1, \quad B_{-1}(x) = 0, \quad R_{-1}(x) = x^{2t} \quad (106)$$

$$A_0(x) = 0, \quad B_0(x) = 1, \quad R_0(x) = T(x) \quad (107)$$

此處，與第(79)式～第(81)式一樣依序所構成的($A_k(x)$, $B_k(x)$, $R_k(x)$)，可以利用與第九實施例中第(82)式相同的歸納法，證明其滿足下面第(108)式。

$$A_k(x)x^{2t} + B_k(x)T(x) = R_k(x) \quad (k \geq -1) \quad (108)$$

另外，第(83)式和第(84)式也同樣成立。雖然根據上述的第(108)式，可知($B_k(x)$, $R_k(x)$)是錯誤位置多項式和錯誤消失數值多項式的候補解，但是以下則證明對於適當的 k 值，提供真正錯誤位置多項式和錯誤消失數值多項式的方式。

換言之，因為 $R_k(x)$ 的次數為單調下降，所以存在一個 k 值，滿足下面第(109)式所示之不等式。

$$\deg R_{k-1} \geq [(2t+h+1)/2] \text{ 且 } \deg R_k < [(2t+h+1)/2] \quad (109)$$

此時，根據第(86)式可知，相關於 B_k 的下面第(110)式會成立。

$$\deg B_k = 2t - \deg R_{k-1} \leq [(2t-h)/2] \quad (110)$$

第(108)式對此 k 成立，而在將 $B_k(x)$ 乘上第(105)式，並且將 $\sigma(x)$ 乘上第(108)式之後，兩邊相加，可得到下面所示之第(111)式。

$$(B_k(x)\varphi(x) + \sigma(x)A_k(x))x^{2t} = B_k(x)\pi(x) + \sigma(x)R_k(x) \quad (111)$$

(請先閱讀背面之注意事項再填寫本頁)

訂

五、發明說明 (58)

另外，根據上述條件 3 以及第(109)式和第(110)式，第(111)式右邊為 $2t-1$ 次數以下的多項式，而如果()內不是零時則其左邊為 $2t$ 以上的多項式，所以可以推論出 $()=0$ 。藉此，與第九實施例一樣，利用上述條件 4 和第(84)式， $(B_k(x), R_k(x))$ 即為 $(\sigma(x), \pi(x))$ 的整數倍。因此，當 $\sigma(x)=B_k(x)/B_k(0)$ ， $\pi(x)=R_k(x)/R_k(0)$ ，可以得到所需要的解。

另外，歐氏演算法的初始值則利用下面第(112)式和第(113)式的方式加以設定。

$$B_{-1}(x) = 0, \quad R_{-1}(x) = x^{2t} \quad (112)$$

$$B_0(x) = \Lambda(x), \quad R_0(x) = T(x) \quad (113)$$

此處，依序構成 $(B_k(x), R_k(x))$ ，並在 $R_k(x)$ 的次數未滿 $[(2t+h+1)/2]$ 時結束歐氏演算法，置換 $\rho(x)=B_k(x)/B_k(0)$ ， $\pi(x)=R_k(x)/R_k(0)$ ，而 $\rho(x)$ 等於 $\sigma(x)\Lambda(x)$ ；對應於消失位置之根以外的根，則是對應於隨機錯誤位置的集合 L 。以下稱 $\rho(x)$ 為錯誤消失位置多項式。另外，錯誤消失位置多項式 $\rho(x)$ 的次數以及錯誤消失數值多項式 $\pi(x)$ 的次數，滿足 $\deg\pi \leq \deg\rho$ 的關係。而在延伸成分中存在錯誤時，等號成立。

在消失位置上的錯誤大小 $e_{m[k]}(k=0, 1, \dots, h-1)$ 能夠利用下面第(114)式求得，另外，在延伸成分以外隨機錯誤位置上的錯誤大小 $e_{l[k]}(k=0, 1, \dots, s-1)$ 能夠利用下面第(115)式求得。而 ρ' 為 ρ 的形式微分。

(請先閱讀背面之注意事項再填寫本頁)

訂

五、發明說明 (59)

$$e_{m[k]} = \frac{\pi(\alpha^{m[k]})}{\rho'(\alpha^{m[k]})\alpha^{(b+2t)m[k]}} \quad (114)$$

$$e_{l[k]} = \frac{\pi(\alpha^{l[k]})}{\rho'(\alpha^{l[k]})\alpha^{(b+2t)l[k]}} \quad (115)$$

而當錯誤位置多項式 $\rho(x)$ 的次數和錯誤數值多項式 $\pi(x)$ 的次數一致時，可知在延伸成分上存在錯誤。而此錯誤的大小則能夠以 $\rho(x)$ 最高次項的係數除以 $\pi(x)$ 最高次項的係數之方式而求得。

此處，第 18 圖表示第十實施例中一階延伸 RS 碼之錯誤訂正方法處理程序之流程圖，上述消失/錯誤訂正方法所適用的情況，是針對 $b=1$ ，最小距離為 7 之高氏場 $GF(2^7)$ 上 $(128, 122)$ 一階延伸 RS 碼。而對於與第九實施例中之對應步驟，則附上與第 16 圖相同的標記，並省略其說明。

圖中，ST27 是藉由伴隨於收訊字組之消失旗標，產生消失位置係數之消失位置係數產生步驟。ST28 則是利用消失位置係數產生步驟 ST27 所產生之消失位置係數，產生消失位置多項式之消失位置多項式產生步驟。ST29 則為修正徵候值多項式產生步驟，是將徵候值多項式產生步驟 ST22 所產生的徵候值多項式和消失位置多項式產生步驟 ST28 所產生之消失位置多項式，兩者相乘，產生修正徵候值多項式。

而歐氏演算法計算步驟 ST23 則與第九實施例之第 16 圖上附加相同標記者不同，其差異點在於是以修正徵

(請先閱讀背面之注意事項再填寫本頁)

訂

五、發明說明 (60)

候值多項式產生步驟 ST29 所產生的修正徵候值多項式為初始值，來計算錯誤消失位置多項式以及錯誤消失數值多項式。另外，比較步驟 ST24 和 Chien 搜尋步驟 ST25 亦分別與第九實施例之第 16 圖上附加相同標記者不同，比較步驟 ST24 的差異點在於其比較錯誤消失位置多項式的次數和錯誤消失數值多項式的次數，Chien 搜尋步驟 ST25 的差異點則在於利用歐氏演算法計算步驟 ST23 所計算出的錯誤消失位置多項式和錯誤消失數值多項式，來計算錯誤位置和錯誤的大小。

接著說明其動作。

在徵候值產生步驟 ST21 中，藉由第(6)式，以所接收的收訊字組來計算徵候值 $S_i(i=0, 1, \dots, 5)$ 。接著在徵候值多項式產生步驟 ST22 中，則是將屬於徵候值 S_i 中延伸成分的徵候值 S_5 設為常數項，根據第(96)式產生徵候值多項式。另外，在消失位置係數產生步驟 ST27 中，利用伴隨收訊字組的消失旗標，產生消失位置係數。接著進入消失位置多項式產生步驟 ST28，當消失的數量 h 在 6 以下時，則利用第(101)式產生消失位置多項式 $\Lambda(x)$ ；當 h 在 7 以上時，則不進行消失的修正，設 $\Lambda(x)=1$ ，進行在第九實施例中所述的解碼動作。此處，當消失的數量在 6 以下時，則能夠訂正到 $(6-h)/2$ 個隨機錯誤。例如 $h=3$ 時，能夠訂正到 3/2 個錯誤，亦即最多訂正一個內部碼所產生的錯誤和延伸成分中所產生的錯誤。

接著在修正徵候值多項式產生步驟 ST29 中，根據

(請先閱讀背面之注意事項再填寫本頁)

不

訂

五、發明說明 (61)

在徵候值多項式產生步驟 ST22 所產生的徵候值多項式 $S(x)$ ，以及在消失位置多項式產生步驟 ST28 所產生的消失位置多項式 $\Lambda(x)$ ，利用下面的第(116)式，計算修正徵候值多項式 $T(x)$ 。

$$T(x) = \Lambda(x)S(x) \quad \text{mod } x^6 \quad (116)$$

接著進入歐氏演算法計算步驟 ST23 中，利用此修正徵候值多項式 $T(x)$ ，實施下面的歐氏演算法。第 19 圖是表示此歐氏演算法計算步驟 ST23 之詳細處理程序的流程圖。如第 19 圖所示，首先利用下面的第(117)式和第(118)式，設定歐氏演算法的初始值(步驟 ST220)。

$$B_{-1}(x) = 0, \quad B_0(x) = \Lambda(x) \quad (117)$$

$$R_{-1}(x) = x^6, \quad R_0(x) = T(x) \quad (118)$$

接著在將變數 i 設定為初始值 0 之後(步驟 ST221)，比較 $R_i(x)$ 的次數和 $[(7+h)/2]$ (步驟 ST222)。在 $R_i(x)$ 的次數尚不到 $[(7+h)/2]$ 之前，遞增變數 i (步驟 ST223)，並且以 $R_{i-1}(x)$ 除以 $R_{i-2}(x)$ 時的商多項式設為 $Q_i(x)$ (步驟 ST224)，藉由第(99)式和第(100)式依序地計算 $(B_i(x), R_i(x))$ (步驟 ST225)。

當 $R_i(x)$ 的次數低於 $[(7+h)/2]$ 時，則設 $(\rho(x)=B_i(x)/B_i(0), \pi(x)=R_i(x)/B_i(x))$ (步驟 ST226)，而終止此歐氏演算法計算步驟 ST23。

接著在比較步驟 ST24 中，比較在上述歐氏演算法計算步驟 ST23 中所得之錯誤消失位置多項式 $\rho(x)$ 之次數和錯誤消失數值多項式 $\pi(x)$ 之次數。若其結果顯示 $\deg\rho \geq$

(請先閱讀背面之注意事項再填寫本頁)

訂

五、發明說明 (62)

$\deg\pi$ 不成立，則做為錯誤的檢測並輸出原來的收訊字組。

另外若 $\deg p \geq \deg\pi$ 成立，在 Chien 搜尋步驟 ST25 中，對錯誤消失位置多項式 $p(x)$ 和錯誤消失數值多項式 $\pi(x)$ 進行 Chien 搜尋以計算錯誤位置，並且根據第(114)式和第(115)式計算各位置上的錯誤大小。

接著進入錯誤訂正步驟 ST26，當在 Chien 搜尋步驟 ST25 中檢查出錯誤個數和錯誤消失位置多項式 $p(x)$ 的次數一致時，則訂正其錯誤並且輸出訂正後的結果。另一方面，當所檢查出的錯誤個數和錯誤消失位置多項式 $p(x)$ 的次數不一致時，則僅止於錯誤的檢查並且輸出原來的收訊字組。

藉此，在第十實施例中，能夠以一次歐氏演算法之計算來進行一階延伸 RS 碼之消失/錯誤訂正，另外，藉由比較由歐氏演算法計算所檢測之錯誤消失位置多項式和錯誤消失數值多項式的次數，也能夠輕易地判斷出是否在延伸成分中產生了錯誤。

【第十一實施例】

在上述第九實施例和第十實施例中，雖然是針對一階延伸 RS 碼的錯誤訂正方法來做說明，但是亦能夠據此實現錯誤訂正裝置。第 20 圖即表示在第十一實施例中具有如此特性之一階延伸 RS 碼的錯誤訂正裝置的方塊圖。圖中，10 是由收訊字組產生徵候值的徵候值產生裝置。11 是徵候值多項式產生裝置，根據徵候值產生裝置

(請先閱讀背面之注意事項再填寫本頁)

訂

五、發明說明 (63)

10 所計算的徵候值，以這些徵候值中包含延伸成分錯誤資訊之徵候值做為常數項位置的方式，產生做為歐氏演算法初始值之徵候值多項式。

12 是歐氏演算法計算裝置，其以徵候值多項式產生裝置 11 所產生的徵候值多項式做為初始值，利用歐氏演算法，計算錯誤位置多項式和錯誤數值多項式。13 是 Chien 搜尋裝置，其對於歐氏演算法計算裝置所計算出的錯誤位置多項式和錯誤數值多項式，進行 Chien 搜尋，用以計算錯誤位置和錯誤數值。14 是用來儲存輸入之收訊字組的收訊字組儲存裝置，15 是用來儲存在 Chien 搜尋裝置 13 中所計算之錯誤位置和錯誤數值的計算結果儲存裝置。16 是根據在計算結果儲存裝置中所儲存的錯誤位置和錯誤數值，用來訂正儲存在收訊字組儲存裝置 14 中收訊字組錯誤的錯誤訂正裝置。

接著說明其動作。

另外此處是利用 $b=1$ ，最小距離為 7 之高氏場 $GF(2^7)$ 上 $(128, 122)$ 的一階延伸 RS 碼，來做說明。

輸入的收訊字組被送到徵候值產生裝置 10 和收訊字組儲存裝置 14。當徵候值產生裝置 10 接收了收訊字組，則計算此收訊字組的徵候值 $S_0, S_1, S_2, S_3, S_4, S_5$ ，並且送到徵候值多項式產生裝置 11 中。另外在收訊字組儲存裝置 5 中則儲存輸入之收訊字組的原來狀態。在徵候值多項式產生裝置 11 中，則是將徵候值產生裝置 10 中所產生的徵候值裡，會受收延伸成分所影響的徵候值

(請先閱讀背面之注意事項再填寫本頁)

訂

五、發明說明 (64)

S_5 設定為常數項，產生如第(96)式所示之微候值多項式。

此微候值多項式產生裝置 11 所產生的微候值多項式被送到歐氏演算法計算裝置 12，而歐氏演算法計算裝置 12 則利用此微候值多項式，進行歐氏演算法的計算，以得到錯誤位置多項式 $\sigma(x)$ 和錯誤數值多項式 $\omega(x)$ 。此處，比較此錯誤位置多項式 $\sigma(x)$ 的次數和錯誤數值多項式 $\omega(x)$ 的次數，如果 $\deg\omega \leq \deg\sigma$ 不成立，則做為錯誤的檢測並輸出原來的收訊字組。另一方面，如果此條件成立時，則將此錯誤位置多項式 $\sigma(x)$ 和錯誤數值多項式 $\omega(x)$ 送到 Chien 搜尋裝置 13，在 Chien 搜尋裝置 13 中計算錯誤位置和錯誤數值。另外，所計算出的錯誤位置和錯誤數值則一直儲存在計算結果儲存裝置 15 中。

當 Chien 搜尋裝置 13 中所檢測出的錯誤個數和錯誤位置多項式 $\sigma(x)$ 的次數相同時，錯誤訂正裝置 16 則讀出以 Chien 搜尋裝置 13 所計算並且儲存在計算結果儲存裝置 15 中所儲存的錯誤位置和錯誤數值，再由收訊字組儲存裝置 14 中所儲存的收訊字組中對應於錯誤位置者，減去其錯誤數值，進行錯誤的訂正，再輸出訂正後的收訊字組。另一方面，當所檢測出的錯誤個數與錯誤位置多項式 $\sigma(x)$ 的次數不一致時，則僅進行錯誤的檢測，並輸出收訊字組儲存裝置 14 中的原來收訊字組。

藉此，在使用習知歐氏演算法之一階延伸 RS 碼的錯誤訂正裝置中，在最差情況下必須進行兩次的歐氏演算法，所以存在解碼延遲的問題，但在第十一實施例中，

(請先閱讀背面之注意事項再填寫本頁)

訂

五、發明說明 (65)

不僅只進行一次的歐氏演算法計算，另外，只需要略為修改習知的 RS 碼錯誤訂正裝置，便可以構成一階延伸 RS 碼錯誤訂正裝置，所以可以得到減少成本的效果。

(請先閱讀背面之注意事項再填寫本頁)

訂

【第十二實施例】

在上述的第十一實施例中所說明的一階延伸 RS 碼錯誤訂正裝置，雖然是以歐氏演算法進行一階延伸 RS 碼的錯誤訂正，但是也能夠適用在用來進行一階延伸 RS 碼之消失/錯誤訂正的一階延伸 RS 碼錯誤訂正裝置。第 21 圖即表示在本發明第十二實施例中具有如此功能的一階延伸 RS 碼錯誤訂正裝置的方塊圖。圖中，17 是藉由伴隨在輸入收訊字組之消失旗標，產生消失位置係數的消失位置係數產生裝置。18 則是以消失位置係數產生裝置 27 所產生的消失位置係數，產生消失位置多項式的消失位置多項式產生裝置。19 則是將此消失位置多項式產生裝置 18 所產生的消失位置多項式和徵候值多項式產生裝置 11 所產生的徵候值多項式兩者相乘，用以產生修正徵候值多項式的修正徵候值多項式產生裝置。

至於其他與第 20 圖中附加相同標記的部分，則是在第十一實施例中者對應的部分，除了歐氏演算法計算裝置 12 和 Chien 搜尋裝置 13 則分別與第 20 圖附加相同的標記，但與第十一實施例中對應者不同，歐氏演算法計算裝置 12 的差異點在於係以修正徵候值多項式產生裝置 19 所產生的修正徵候值多項式，做為初始值，以計算

五、發明說明 (66)

錯誤消失位置多項式和錯誤消失數值多項式。Chien 搜尋裝置 13 的差異點在於係根據歐氏演算法計算裝置 12 所計算出的錯誤消失位置多項式和錯誤消失數值多項式，來計算錯誤位置和其錯誤的大小。

接著說明其動作。

另外此處是利用 $b=1$ ，最小距離為 7 之高氏場 $GF(2^7)$ 上 $(128, 122)$ 的一階延伸 RS 碼，來做說明。

首先，輸入的收訊字組分別送到徵候值產生裝置 10，消失位置係數產生裝置 17 和收訊字組儲存裝置 14。與第十一實施例中的情況相同，在徵候值產生裝置 10 中，計算收訊字組的徵候值 $S_0, S_1, S_2, S_3, S_4, S_5$ ，收訊字組儲存裝置 14 中則儲存原來的收訊字組。而在徵候值多項式產生裝置 11 中，則是將徵候值產生裝置 10 中所產生的徵候值裡，會受收延伸成分所影響的徵候值 S_5 設定為常數項，產生如第(96)式所示之徵候值多項式。

另一方面，接收收訊字組的消失位置係數產生裝置 17，則根據伴隨在消失旗標，產生消失位置係數，並將其送到消失位置多項式產生裝置 18。在消失位置多項式產生裝置 18 中，則利用此消失位置係數產生裝置 17 所產生的消失位置係數，產生消失位置多項式，並將其送到修正徵候值多項式產生裝置 19。修正徵候值多項式產生裝置 19 則將徵候值多項式產生裝置 11 所產生的徵候值多項式，以及在此消失位置多項式產生裝置 18 中所產生的消失位置多項式，兩者相乘，產生修正徵候值多項

(請先閱讀背面之注意事項再填寫本頁)

訂

五、發明說明 (67)

式，並將其送到歐氏演算法計算裝置 12。

在歐氏演算法計算裝置 12 中，則是利用從修正徵候值多項式產生裝置 19 所取得的修正徵候值多項式，進行歐氏演算法的計算，以計算錯誤消失位置多項式 $\rho(x)$ 和錯誤消失數值多項式 $\pi(x)$ 。此處，比較此錯誤消失位置多項式 $\rho(x)$ 的次數和錯誤消失數值多項式 $\pi(x)$ 的次數。如果 $\deg\pi \leq \deg\rho$ 不成立，則進行錯誤的檢測，並輸出原來的收訊字組。另外，如果此條件成立，則在 Chien 搜尋裝置 13 中，藉由這些錯誤消失位置多項式 $\rho(x)$ 和錯誤消失數值多項式 $\pi(x)$ ，來計算錯誤位置和錯誤數值，並將其送到計算結果儲存裝置 15 中加以儲存。

當 Chien 搜尋裝置 13 中所檢測出的錯誤個數和錯誤消失位置多項式 $\rho(x)$ 的次數相同時，錯誤訂正裝置 16 則從收訊字組儲存裝置 14 所儲存的收訊字組中，對應於以 Chien 搜尋裝置 13 計算並儲存在計算結果儲存裝置 15 中的錯誤位置，減去其錯誤數值，進行錯誤訂正，並輸出訂正後的收訊字組。另一方面，當所檢測出的錯誤個數與錯誤消失位置多項式 $\rho(x)$ 的次數不一致時，則僅進行錯誤的檢測，並輸出收訊字組儲存裝置 14 中的原來收訊字組。

藉此，在第十二實施例中，能夠以一次的歐氏演算法計算來進行一階延伸 RS 碼的消失/錯誤訂正，並且能夠對歐氏演算法計算所檢測出的錯誤消失位置多項式和錯誤消失數值多項式，藉由比較其次數的方式，輕易地

(請先閱讀背面之注意事項再填寫本頁)

訂

五、發明說明 (68)

判斷出是否延伸成分中產生了錯誤。另外，只需要略為修改習知的 RS 碼錯誤訂正裝置，便可以構成一階延伸 RS 碼錯誤訂正裝置，所以可以得到減少成本的效果。

【第十三實施例】

接著，本發明第十三實施例中，則是對於利用 Berlekamp-Massey 演算法的二階延伸 RS 碼錯誤訂正方法加以說明。

在第十實施例的說明中，是採用習知技術中說明的碼長 n 、資訊符號數 k 、最小距離 $d=n-k+1=2t+1$ (t 為正整數)之二階延伸 RS 碼。

第 22 圖表示本發明第十三實施例中二階延伸 RS 碼之錯誤訂正方法處理程序之流程圖。圖中，ST31 是以輸入的收訊字組來計算其徵候值的徵候值產生步驟，ST32 為 Berlekamp-Massey 演算法計算步驟，其採用徵候值產生步驟 ST31 所計算的徵候值，進行 Berlekamp-Massey 演算法，並且藉由形式移位暫存器長度 L 的值來控制是否使用包含第二延伸成分錯誤資訊的徵候值，以依序計算錯誤位置多項式。

另外，ST33 則是 Chien 搜尋步驟，對於此 Berlekamp-Massey 演算法計算步驟 ST32 中所計算出的錯誤位置多項式，進行 Chien 搜尋，以計算出錯誤位置多項式的根(錯誤位置)。ST34 則是判斷在 Chien 搜尋步驟 ST33 所得到的錯誤位置數量是否適當的檢定步驟，ST35 則是錯誤訂正步驟，如果在檢定步驟 ST34 的判斷

(請先閱讀背面之注意事項再填寫本頁)

訂

五、發明說明 (69)

結果顯示 Chien 搜尋步驟 ST33 所得到的錯誤位置數量是適當時，則使用基於徵候值產生步驟 ST31 所計算出的徵候值而產生的徵候值多項式，計算出錯誤數值，以訂正收訊字組的錯誤。

接著說明其動作。

而在以下的說明中，係以收訊字組 $R = (r_{n-2}, r_{n-3}, \dots, r_0, r_1)$ 來做說明。首先在徵候值產生步驟 ST31 中，利用上述的第(25)式，以輸入的收訊字組來計算徵候值 $S = (S_0, S_1, \dots, S_{2t-2}, S_{2t-1})$ 。此處在上述第(25)式中的 H_2 則是第(24)式所提供的配類陣列。

第 23 圖表示此 Berlekamp-Massey 演算法計算步驟 ST32 的具體程序之流程圖。以下參考此第 23 圖，以說明此 Berlekamp-Massey 演算法。

如第 23 圖所示，首先，將用來計數步驟數量的變數 r 設為 1，錯誤位置多項式的初始值 $\sigma^{(0)}(x)$ 設為 1，形式移位暫存器長度 L 設為 0，用來更新錯誤位置多項式的輔助多項式 $\tau^{(0)}(x)$ 設為 1(步驟 ST310)。

依序計算錯誤位置多項式，而在第 $r-1$ 次步驟結束後所計算出者，則為錯誤位置多項式 $\sigma^{(r-1)}(x)$ 和輔助多項式 $\tau^{(r-1)}(x)$ 。在第 r 次步驟中，利用 $\sigma^{(r-1)}(x)$ 的係數，計算如第(35)式所示之偏差值 Δ_r (步驟 ST311)。接著，檢查其計算結果(ST312)，如果偏差值 Δ_r 為 0，則不變更錯誤位置多項式，第 r 次的錯誤位置多項式即為第 $r-1$ 次所得到的 $\sigma^{(r-1)}(x)$ (ST313)。另一方面，如果偏差值 Δ_r 不為 0

(請先閱讀背面之注意事項再填寫本頁)

訂

五、發明說明 (70)

時，則使用輔助多項式 $\tau^{(r-1)}(x)$ ，藉由第(27)式構成接續多項式 $\eta(x)$ (步驟 ST314)。

接著，檢查形式移位暫存器長度 L 是否滿足 $2L < r$ 的關係(步驟 ST315)，當滿足時，則利用第(28)式~第(30)式所示之方式，更新輔助多項式 $\tau^{(r)}(x)$ ，錯誤位置多項式 $\sigma^{(r)}(x)$ 和形式移位暫存器長度 L (步驟 ST316)。另外當此不等式不滿足時，則不更新形式移位暫存器長度 L ，以第(31)式更新錯誤位置多項式 $\sigma^{(r)}(x)$ (步驟 ST317)，並以第(32)式更新輔助多項式 $\tau^{(r)}(x)$ (步驟 ST313)。

此處，若錯誤個數在 $t-1$ 個以下，則在第 $r=2t-1$ 次步驟所計算出的偏差值 Δ_{2t-1} 為 0，形式移位暫存器長度 L 在 $t-1$ 以下。但當 $L \leq t-1$ 成立時，則 $\Delta_{2t-1}=0$ 自動成立。此可以利用矛盾法加以證明。

亦即，假設 $\Delta_{2t-1} \neq 0$ 而導出矛盾的情況。當在步驟 $2t-2, 2t-1$ 中的形式移位暫存器長度為 $L_{2t-2}, L_{2t-1}(=L)$ 時，藉由上述的假設 $\Delta_{2t-1} \neq 0$ ，則 $L_{2t-1} = \max(L_{2t-2}, 2t-1-L_{2t-2})$ 。而因為右側大於 $t-1$ ，所以能夠得到 $L=L_{2t-1} > t-1$ 的結論。此與 $L \leq t-1$ 的假定相反。因此證明了 $\Delta_{2t-1}=0$ 。

藉由上述的觀察，當形式移位暫存器長度 L 滿足下面所示第(119)式的條件時，即是在內部碼和第一延伸成分中產生了 $t-1$ 個以下的錯誤，並將 $\sigma(x)=\sigma^{(2t-1)}(x)$ ，結束此 Berlekamp-Massey 演算法計算步驟 ST32。

$$L \leq t-1$$

(119)

而當上述第(119)式的條件不滿足時，即是在內部

(請先閱讀背面之注意事項再填寫本頁)

訂

五、發明說明 (71)

碼和第一延伸成分中產生了 t 個以上的錯誤，利用 S_{2t-1} 再進行一步驟的 Berlekamp-Massey 演算法，並將 $\sigma(x) = \sigma^{(2t)}(x)$ ，結束此 Berlekamp-Massey 演算法計算步驟 ST32。

亦即，檢查用來計數步驟數量的變數 r 是否為 $2t$ (步驟 ST318)，如果變數不為 $2t$ ，則檢查變數 r 是否達到 $2t-1$ ，以及形式移位暫存器以外長度 L 是否滿足上述第(119)式所示之條件(步驟 ST319)。其結果若顯示無法滿足這兩個條件時，則遞增變數 r (步驟 ST320)，同時返回步驟 ST311 以下的處理。另一方面，若變數 r 達到 $2t-1$ 並且形式移位暫存器長度 L 滿足第(119)式的條件時，則將 $\sigma(x) = \sigma^{(2t-1)}(x)$ ，結束此 Berlekamp-Massey 演算法計算步驟 ST32。另外在步驟 ST318 中，當檢查出變數 r 為 $2t$ 時，則將 $\sigma(x) = \sigma^{(2t)}(x)$ ，結束此 Berlekamp-Massey 演算法計算步驟 ST32。

在此 Berlekamp-Massey 演算法計算步驟 ST32 中所計算出的錯誤位置多項式 $\sigma(x)$ ，滿足下面所示的第(120)式。若在第一延伸成分沒有產生錯誤時，則等號成立，而在第一延伸成分產生錯誤時，錯誤位置多項式 $\sigma(x)$ 的次數會比形式移位暫存器長度 L 來得小。

$$\deg \sigma \leq L \quad (120)$$

接著在 Chien 搜尋步驟 ST33 中，對於上述 Berlekamp-Massey 演算法計算步驟 ST32 所計算出的錯誤位置多項式 $\sigma(x)$ 進行 Chien 搜尋，以計算出錯誤位置

(請先閱讀背面之注意事項再填寫本頁)

訂

五、發明說明 (72)

多項式 $\sigma(x)$ 的根(錯誤位置)。接著在檢定步驟 ST34 中，判斷在 Chien 搜尋步驟 ST33 所獲得的錯誤位置多項式 $\sigma(x)$ 的根數量，是否與錯誤位置多項式 $\sigma(x)$ 的次數一致。其結果若顯示計算出的根數量與錯誤位置多項式 $\sigma(x)$ 的次數一致時，在錯誤訂正步驟 ST35 中計算出錯誤數值，而在收訊字組的對應錯誤位置上減去此錯誤數值，進行錯誤的訂正，並輸出訂正後的收訊字組。另外，當所計算出的根數量與錯誤位置多項式 $\sigma(x)$ 的次數不一致時，則只進行錯誤檢測，並將原來的收訊字組連同錯誤檢出旗標一併輸出。

在此錯誤訂正步驟 ST35 中計算錯誤數值時，是利用以下所示的第(121)式，計算錯誤數值多項式 $\omega(x)$ 。

$$\omega(x) \equiv \sigma(x)S(x) \pmod{x^{2t-1}} \quad (121)$$

此處，藉由構成錯誤位置多項式 $\sigma(x)$ 的方式，錯誤數值多項式 $\omega(x)$ 的 L 次以上項為 0。因此，只需要計算 $L-1$ 次以下的項即可。而在上述錯誤數值多項式 $\omega(x)$ 的計算，則是使用下面第(122)所示的微候值多項式 $S(x)$ 。

$$S(x) = S_0 + S_1x + \dots + S_{2t-2}x^{2t-2} \quad (122)$$

藉此，各錯誤位置 j (對應於錯誤位置多項式的根 α^{-j})中的錯誤數值 e_j ，則與第九實施例的情況相同，能夠利用下面所示之第(123)式求得。

$$e_j = \frac{\omega(\alpha^{-j})}{\sigma'(\alpha^{-j})\alpha^{-j}} \quad (123)$$

另外，在訂正內部碼所產生的錯誤上，也可以使用

(請先閱讀背面之注意事項再填寫本頁)

訂

五、發明說明 (73)

以下第(124)式所示的徵候值多項式。

$$S(x) = S_0 + S_1 x + \cdots + S_{2t-3} x^{2t-3} \quad (124)$$

此時，錯誤數值多項式 $\omega(x)$ 即表示如下面之第(125)式。

$$\omega(x) \equiv \sigma(x)S(x) \pmod{x^{2t-2}} \quad (125)$$

在此情況下，各錯誤位置 j (對應於錯誤位置多項式的根 α^{-j})中的錯誤數值 e_j ，也則與第九實施例的情況相同，能夠利用下面所示之第(126)式求得。

$$e_j = \frac{\omega(\alpha^{-j})}{\sigma'(\alpha^{-j})} \quad (126)$$

接著，在此第九實施例中，係採用碼長 $n=17$ 、資訊符號數 $k=11$ 、最小距離 $7(t=3)$ 之高氏場 $GF(16)$ 上的二階延伸 RS 碼，來具體說明。此處，原始多項式設為 x^4+x+1 ，其根設為 α 。 α 是原始第 15 根(primitive fifteenth roots)。此二階延伸 RS 碼的配類陣列則是由下述的第(127)式所提供之。

$$H_2 = \begin{bmatrix} 0 & 1 & 1 & \cdots & 1 & 1 & 1 \\ 0 & \alpha^{14} & \alpha^{13} & \cdots & \alpha & 1 & 0 \\ 0 & \alpha^{2 \times 14} & \alpha^{2 \times 13} & \cdots & \alpha^2 & 1 & 0 \\ 0 & \alpha^{3 \times 14} & \alpha^{3 \times 13} & \cdots & \alpha^3 & 1 & 0 \\ 0 & \alpha^{4 \times 14} & \alpha^{4 \times 13} & \cdots & \alpha^4 & 1 & 0 \\ 1 & \alpha^{5 \times 14} & \alpha^{5 \times 13} & \cdots & \alpha^5 & 1 & 0 \end{bmatrix} \quad (127)$$

現以收訊字組 $R=(r_{15}, r_{14}, \dots, r_0, r_{-1})$ 為所收訊號，在第一延伸成分、第 3 成分、第 10 成分上則發生 α^{14} 、 α^6 、 α^5 的錯誤。此時，利用第(25)式，徵候值 $S=(S_0, S_1, S_2, S_3, S_4, S_5)$ 則計算為： $S_0=\alpha^4$ ， $S_1=\alpha^7$ ， $S_2=\alpha^3$ ， $S_3=\alpha^{10}$ ， $S_4=\alpha^{14}$ ，

(請先閱讀背面之注意事項再填寫本頁)

訂

五、發明說明 (74)

$$S_5 = \alpha^7.$$

如第 23 圖所示，錯誤位置多項式 $\sigma^{(0)}(x)$ 設為 1，形式移位暫存器長度 L 設為 0，用來更新錯誤位置多項式的輔助多項式 $\tau^{(0)}(x)$ 設為 1，計算步驟數量的變數 r 設為 1。首先，利用 S_0, S_1, S_2, S_3, S_4 的 5 個徵候值，實施 Berlekamp-Massey 演算法的計算。

在 $r=1$ ，偏差值 $\Delta_1 = \alpha^4$ ，另外，因為 $2L=0 < r=1$ ，所以 $L=r-L=1$ 。錯誤位置多項式 $\sigma^{(1)}$ 和輔助多項式 $\tau^{(1)}$ 則以下面所示的第(128)式和第(129)式的方式求得。

$$\sigma^{(1)}(x) = \alpha^4 x + 1 \quad (128)$$

$$\tau^{(1)}(x) = \alpha^{11} \quad (129)$$

接著，在 $r=2$ ，偏差值 $\Delta_2 = \alpha^{11}$ ，另外，因為 $2L=2, r=2$ ，所以 $L=1$ 。錯誤位置多項式 $\sigma^{(2)}$ 和輔助多項式 $\tau^{(2)}$ 則以下面所示的第(130)式和第(131)式的方式求得。

$$\sigma^{(2)}(x) = \alpha^3 x + 1 \quad (130)$$

$$\tau^{(2)}(x) = \alpha^{11} x \quad (131)$$

接著，在 $r=3$ ，偏差值 $\Delta_3 = \alpha^{12}$ ，另外，因為 $2L=2 < r=3$ ，所以 $L=r-L=2$ 。錯誤位置多項式 $\sigma^{(3)}$ 和輔助多項式 $\tau^{(3)}$ 則以下面所示的第(132)式和第(133)式的方式求得。

$$\sigma^{(3)}(x) = \alpha^8 x^2 + \alpha^3 x + 1 \quad (132)$$

$$\tau^{(3)}(x) = \alpha^6 x + \alpha^3 \quad (133)$$

接著，在 $r=4$ ，偏差值 $\Delta_4 = \alpha^9$ ，另外，因為 $2L=4, r=4$ ，所以 $L=2$ 。錯誤位置多項式 $\sigma^{(4)}$ 和輔助多項式 $\tau^{(4)}$ 則以下面所示的第(134)式和第(135)式的方式求得。

(請先閱讀背面之注意事項再填寫本頁)

訂

五、發明說明 (75)

$$\sigma^{(4)}(x) = \alpha^2 x^2 + \alpha^{10} x + 1 \quad (134)$$

$$\tau^{(4)}(x) = \alpha^6 x^2 + \alpha^3 x \quad (135)$$

接著，在 $r=5$ ，偏差值 $\Delta_5 = \alpha^{14}$ ，另外，因為 $2L=4 < r=5$ ，所以 $L=r-L=3$ 。錯誤位置多項式 $\sigma^{(5)}$ 和輔助多項式 $\tau^{(5)}$ 則以下面所示的第(136)式和第(137)式的方式求得。

$$\sigma^{(5)}(x) = \alpha^5 x^3 + \alpha^{10} x + 1 \quad (136)$$

$$\tau^{(5)}(x) = \alpha^3 x^2 + \alpha^{11} x + \alpha \quad (137)$$

而在此 $r=5$ 中，因為 $L=3 > t-1=2$ ，所以遞增 r 並且再實施一次 Berlekamp-Massey 演算法計算。

在 $r=6$ 時，偏差值 $\Delta_6 = \alpha^2$ ，另外，因為 $2L=6$ 、 $r=6$ ，所以 $L=3$ 。錯誤位置多項式 $\sigma^{(6)}$ 和輔助多項式 $\tau^{(6)}$ 則以下面所示的第(138)式和第(139)式的方式求得。

$$\sigma^{(6)}(x) = \alpha^{13} x^2 + \alpha^{12} x + 1 \quad (138)$$

$$\tau^{(6)}(x) = \alpha^3 x^3 + \alpha^{11} x^2 + \alpha x \quad (139)$$

而此 $\sigma^{(6)}(x)$ 可以因此分解成 $\sigma^{(6)}(x) = (\alpha^3 x + 1)(\alpha^{10} + 1)$ ，所以求得根 α^{-3} 、 α^{-10} ，得知第 3 成分和第 10 成分上產生了錯誤。另外，雖然錯誤位置多項式的次數是 2 次，但因為形式移位暫存器長度 L 為 3，所以也可以得知第一延伸成分上產生了錯誤。

接著，以下述的第(140)式方式，設定徵候值多項式。

$$S(x) = S_1 + S_2 x + S_2 x^2 + S_2 x^3 = \alpha^7 + \alpha^3 x + \alpha^{10} x^2 + \alpha^{14} x^3 \quad (140)$$

接著使用此徵候值多項式 $S(x)$ ，藉由下面第(141)式，計算錯誤數值多項式。

(請先閱讀背面之注意事項再填寫本頁)

訂

五、發明說明 (76)

$$\begin{aligned}\omega(x) &\equiv \sigma(x)S(x) \pmod{x^4} \\ &\equiv \alpha^4 + \alpha^{14}x + \alpha^{12}x^2\end{aligned}\quad (141)$$

藉此，第 3 成分和第 10 成分中的錯誤數值 e_3, e_{10} ，則藉由下述的第(142)式和第(143)式求得。

$$e_3 = \frac{\omega(\alpha^{-3})}{\sigma'(\alpha^{-3})} = \alpha^6 \quad (142)$$

$$e_{10} = \frac{\omega(\alpha^{-10})}{\sigma'(\alpha^{-10})} = \alpha^5 \quad (143)$$

藉此，在使用習知 Berlekamp-Massey 演算法之二階延伸 RS 碼的錯誤訂正方法中，在最差情況下必須進行兩次的 Berlekamp-Massey 演算法計算，所以存在解碼延遲的問題，但在第十三實施例中，不僅只進行一次的 Berlekamp-Massey 演算法計算，另外，藉由將 Berlekamp-Massey 演算法計算所檢測出的錯誤位置多項式之次數和形式移位暫存器長度進行比較，所以能夠輕易判斷出在第一延伸成分中是否產生錯誤等等的效果。

(請先閱讀背面之注意事項再填寫本頁)

訂

【第十四實施例】

在上述第十三實施例中，雖然是針對以 Berlekamp-Massey 演算法來進行二階延伸 RS 碼錯誤訂正的二階延伸 RS 碼錯誤訂正方法進行說明，但是也可以適用於用來進行二階延伸 RS 碼消失/錯誤訂正的二階延伸 RS 碼錯誤訂正方法。以下即對於本發明第十四實施例，二階延伸 RS 碼之消失/錯誤訂正方法的原理加以說明。而在第十四實施例的說明中，是採用習知技術中說明的碼長

五、發明說明 (77)

n、資訊符號數 k、最小距離 $d=n-k+1=2t+1$ (t 為正整數)
之二階延伸 RS 碼。

第 24 圖表示本發明第十四實施例中，二階延伸 RS 碼之錯誤訂正方法之處理程序的流程圖，與第十三實施例相對應的步驟，則附上與第 22 圖相同的符號，並省略其說明。圖中，ST36 是消失位置多項式產生步驟，利用輸入收訊字組所判隨的消失旗標，產生消失位置係數，再藉由消失位置係數，計算消失位置多項式。而 Berlekamp-Massey 演算法計算步驟 ST32 則與第十三實施例之第 22 圖中附上相同標記者不同，其差異點在於將消失位置多項式產生步驟 ST36 中所計算的消失位置多項式，設定為初始值，再利用微候值產生步驟 ST31 所計算出的微候值，進行 Berlekamp-Massey 演算法，而藉由形式移位暫存器長度的值，控制是否使用包含第二延伸成分錯誤資訊的微候值，以依序計算錯誤位置多項式。

(請先閱讀背面之注意事項再填寫本頁)

訂

接著說明其動作。

在以下說明中，係以收訊字組 $R=(r_{n-2}, r_{n-3}, \dots, r_0, r_1)$ 為所收訊號來做說明。首先，在微候值產生步驟 ST31 中，利用第(25)式計算微候值 $S=(S_0, S_1, \dots, S_{2t-2}, S_{2t-1})$ 。第(25)式中的 H_2 則為第(24)式所提供的配類陣列。

此處，利用伴隨在收訊字組 R 的消失旗標，將判斷出消失的內部碼符號集合設為 M，此集合 M 的元素則為 $m[k](k=0, 1, \dots, h-1)$ 。另外，將上述消失以外的隨機

五、發明說明 (78)

錯誤集合設為 L ，此集合 L 的元素則為 $l[k](k=0, 1, \dots, s-1)$ 。現在假設 $2s+h \leq 2t$ 成立，說明對於消失和隨機錯誤的訂正方式。

換言之，在消失位置多項式產生步驟 ST36 中，利用伴隨在收訊字組 R 之消失旗標而判斷出消失的消失位置集合 M 之各元素 $m[k]$ ，構成消失位置係數 $\alpha^{m[k]}$ ，利用此消失位置係數 $\alpha^{m[k]}$ ，構成下面第(144)式所示之多項式，其稱為消失位置多項式。

$$\Lambda(x) = (1 + \alpha^{m[0]}x)(1 + \alpha^{m[1]}x) \cdots (1 + \alpha^{m[h-1]}x) \quad (144)$$

接著進入 Berlekamp-Massey 演算法計算步驟 ST32，設定此消失位置多項式 $\Lambda(x)$ 為初始值，利用從收訊字組 R 所計算出的 $2t$ 個微候值內，除了 S_{2t-1} 之外的 $S_0, S_1, \dots, S_{2t-2}$ ，進行 Berlekamp-Massey 演算法計算。由於使用了此 $2t-1$ 個微候值，所以能夠訂正 h 個消失和 $t=[(2t-h-1)/2]$ 個隨機錯誤。第 25 圖表示此 Berlekamp-Massey 演算法演算步驟 ST32 的實際程序。

如第 25 圖所示，首先將計數步驟數的變數 r 設為 $h+1$ ，錯誤位置多項式 $\sigma^{(h)}(x)$ 設為 $\Lambda(x)$ ，形式移位暫存器長度 L 設為 h ，而用來更新錯誤位置多項式的輔助多項式 $\tau^{(h)}(x)$ 則設為 $\Lambda(x)$ (步驟 ST330)。

依序計算錯誤位置多項式，而在第 $r-1$ 次步驟結束後所計算出者，則為錯誤位置多項式 $\sigma^{(r-1)}(x)$ 和輔助多項式 $\tau^{(r-1)}(x)$ 。在第 r 次步驟中，利用 $\sigma^{(r-1)}(x)$ 的係數，計算如下面第(145)式所示之偏差值 Δ_r (步驟 ST331)。

(請先閱讀背面之注意事項再填寫本頁)

訂

五、發明說明 (79)

$$\Delta_r = \sum_{j=0}^L \sigma_j^{(r-1)} S_{r-j-1} \quad (145)$$

接著，檢查其計算結果(ST332)，如果偏差值 Δ_r 為 0，則不變更錯誤位置多項式 $\sigma(x)$ ，第 r 次的錯誤位置多項式即為第 $r-1$ 次所得者(ST333)。另一方面，如果偏差值 Δ_r 不為 0 時，則使用輔助多項式 $\tau^{(r-1)}(x)$ ，藉由第(27)式構成接續多項式 $\eta(x)$ (步驟 ST334)。

接著，檢查形式移位暫存器長度 L 是否滿足 $2L < r+h$ (步驟 ST335)，若滿足時，則利用第(146)式~第(148)式所示的方式，更新輔助多項式、錯誤位置多項式和形式移位暫存器長度 L (步驟 ST336)。

$$\tau^{(r)}(x) \leftarrow \Delta_r^{-1} \sigma^{(r-1)}(x) \quad (146)$$

$$\sigma^{(r)}(x) \leftarrow \eta(x) \quad (147)$$

$$L \leftarrow r - L + h \quad (148)$$

另外，若不滿足上述不等式時，則不更新形式移位暫存器長度 L ，以第(149)式的方式更新錯誤位置多項式(步驟 ST337)，再以第(150)式的方式更新輔助多項式(步驟 ST333)。

$$\sigma^{(r)}(x) \leftarrow \eta(x) \quad (149)$$

$$\tau^{(r)}(x) \leftarrow x \tau^{(r-1)}(x) \quad (150)$$

此處，若隨機錯誤的個數在 t_0 個以下，則在 $r=2t-1$ 步驟結束之後所計算出的形式移位暫存器長度 L ，會滿足下面第(151)式所示的條件。

(請先閱讀背面之注意事項再填寫本頁)

訂

五、發明說明 (80)

$$L \leq h + t_0$$

(151)

因此，當形式移位暫存器長度 L 滿足第(151)式的條件時，即推測在內部碼和第一延伸成分中，產生了 h 個消失和 t_0 個以下的隨機錯誤，將 $\sigma(x)=\sigma^{(2t-1)}(x)$ ，結束此 Berlekamp-Massey 演算法計算步驟 ST32。

而當上述第(151)式的條件不滿足時，即表示在內部碼和第一延伸成分上產生了 t_0+1 個以上的隨機錯誤，所以利用 S_{2t-1} 再進行一步驟的 Berlekamp-Massey 演算法，設 $\sigma(x)=\sigma^{(2t)}(x)$ ，結束 Berlekamp-Massey 演算法計算步驟 ST32。

換言之，判斷是否用來計算步驟數的變數 r 變為 $2t$ (步驟 ST338)，若變數 r 不是 $2t$ ，則檢查變數 r 是否達到 $2t-1$ ，並且形式移位暫存器長度 L 是否滿足上述第(151)式所示的條件(步驟 ST339)。其結果若顯示沒有滿足這兩個條件，則遞增變數 r (步驟 ST340)，並且返回步驟 ST331 以下的處理。另一方面，若變數 r 達到 $2t-1$ ，並且形式移位暫存器長度 L 滿足上述第(151)式所示的條件，則 $\sigma(x)=\sigma^{(2t-1)}(x)$ ，結束此 Berlekamp-Massey 演算法計算步驟 ST32。

在此 Berlekamp-Massey 演算法計算步驟 ST32 中所計算出的錯誤位置多項式 $\sigma(x)$ 和形式移位暫存器長度 L ，則滿足下面第(152)式所示之條件。而在第一延伸成分上未產生錯誤時，等號成立，但是在第一延伸成分上產生錯誤時，則錯誤位置多項式 $\sigma(x)$ 的次數小於形式移

(請先閱讀背面之注意事項再填寫本頁)

訂

五、發明說明 (81)

位暫存器長度 L

$$\deg \sigma \leq L$$

(152)

接著在 Chien 步驟 ST33 中，對於在上述 Berlekamp-Massey 算步驟 ST32 中所計算出的錯誤位置多項式 (Chien 搜尋，以計算出錯誤位置多項式 $\sigma(x)$ 之位置)。接著在檢定步驟 ST34 中，判斷在 Chien ST33 中所得到的錯誤位置多項式 $\sigma(x)$ 之根數與錯誤位置多項式 $\sigma(x)$ 的次數一致。其結果與計算出的根數量和錯誤位置多項式 $\sigma(x)$ 的次數則在錯誤訂正步驟 ST35 中計算錯誤數值，而在中的對應錯誤位置上減去其錯誤數值，進行錯並輸出訂正後的收訊字組。另外，當所計算出和錯誤位置多項式 $\sigma(x)$ 的次數不一致時，則僅的檢測，將錯誤檢出旗標連同原來的收訊字組。

在此錯誤訂正 ST35 中的錯誤數值之計算，首先，用徵候值多項錯誤數值多項式 $\omega(x)$ 。此處，利用計算步驟數的以下面的方式，設定徵候值多項式 $S(x)$ 。亦即多項式的設定，在變數 r 為 $2t$ 時是利用下面的式，而在變數 r 為 $2t-1$ 時則是利用第 (154) 式。這當第二延伸成分上產生錯誤時，由於以使用 S_{2t} 無法提供真正的錯誤數值多項式，故避免使用。

$$S(x) = S_0 + S_1 x + \dots +$$

(153)

(請先閱讀背面之注意事項再填寫本頁)

訂

五、發明說明 (82)

$$S(x) = S_0 + S_1x + \cdots + S_{2t-2}x^{2t-2} \quad (154)$$

在使用上述第(153)式時，是以下面所示的第(155)式來計算錯誤數值多項式 $\omega(x)$ ，在使用上述第(154)式時，則是以下面所示的第(156)式來計算錯誤數值多項式 $\omega(x)$ 。

$$\omega(x) = \sigma(x)S(x) \mod x^{2t} \quad (155)$$

$$\omega(x) = \sigma(x)S(x) \mod x^{2t-1} \quad (156)$$

藉此，在各消失/錯誤位置 j 上的錯誤數值 e_j ，與第九實施例的情況相同，能夠利用下面所示的第(157)式求得。

$$e_j = \frac{\omega(\alpha^{-j})}{\sigma'(\alpha^{-j})\alpha^{-j}} \quad (157)$$

接著，在此第十四實施例中，係採用碼長 $n=17$ 、資訊符號數 $k=11$ 、最小距離 $7(t=3)$ 之高氏場 $GF(16)$ 上的二階延伸 RS 碼，來具體說明。而原始方程式和配類陣列則與上述第十三實施例的情況相同。

現以收訊字組 $R=(r_{15}, r_{14}, \dots, r_0, r_{-1})$ 為所收訊號，其中在第一延伸成分、第 5 成分、第 7 成分和第 9 成分上產生 α^8 、 α^{10} 、 α^7 、 α^{11} 的錯誤，並在第 5 和第 7 成分上產生消失旗標。此時，徵候值 $S=(S_0, S_1, S_2, S_3, S_4, S_5)$ 則利用第(25)式計算為： $S_0=\alpha^{10}$ 、 $S_1=\alpha^{11}$ 、 $S_2=\alpha^4$ 、 $S_3=\alpha^{12}$ 、 $S_4=\alpha^4$ 、 $S_5=\alpha^{10}$ 。

另外，由消失位置 5、7 構成消失位置係數 α^5 和 α^7 ，利用此消失位置係數，構成以下面第(158)式所示之消失

(請先閱讀背面之注意事項再填寫本頁)

訂

五、發明說明 (83)

位置多項式 $\Lambda(x)$ 。

$$\Lambda(x) = (1 + \alpha^5 x)(1 + \alpha^5 x) = \alpha^{12} x^2 + \alpha^{13} x + 1 \quad (158)$$

如第 25 圖所示，計算步驟數之變數 r 的初始值設為 $h+1=3$ ，錯誤位置多項式 $\sigma^{(2)}(x)$ 設為 $\Lambda(x)$ ，形式移位暫存器長度 L 設為 $h=2$ ，用來更新錯誤位置多項式的輔助多項式 $\tau^{(2)}(x)$ 設為 $\Lambda(x)$ 。另外，設 $t_0=[(5-h)/2]=1$ 。首先，使用五個徵候值 S_0, S_1, S_2, S_3, S_4 進行 Berlekamp-Massey 演算法計算。

在 $r=3$ ，偏差值 $\Delta_3=\alpha$ ，另外，因為 $2L=4 < r+h=5$ ，所以 $L=r-L+h=3$ 。錯誤位置多項式 $\sigma^{(3)}$ 和輔助多項式 $\tau^{(3)}$ 則以下面所示的第(159)式和第(160)式的方式求得。

$$\sigma^{(3)}(x) = \alpha^{13} x^3 + \alpha^5 x^2 + \alpha^{12} x + 1 \quad (159)$$

$$\tau^{(3)}(x) = \alpha^{11} x^2 + \alpha^{12} x + \alpha^{14} \quad (160)$$

接著，在 $r=4$ ，偏差值 $\Delta_4=\alpha^9$ ，另外，因為 $2L=6$ 、 $r+h=6$ ，所以 $L=3$ 。錯誤位置多項式 $\sigma^{(4)}$ 和輔助多項式 $\tau^{(4)}$ 則以下面所示的第(161)式和第(162)式的方式求得。

$$\sigma^{(4)}(x) = \alpha^7 x^3 + \alpha^9 x^2 + \alpha^9 x + 1 \quad (161)$$

$$\tau^{(4)}(x) = \alpha^{11} x^3 + \alpha^{12} x^2 + \alpha^{14} x \quad (162)$$

接著，在 $r=5$ ，偏差值 $\Delta_5=\alpha^9$ ，另外，因為 $2L=6 < r+h=7$ ，所以 $L=4$ 。錯誤位置多項式 $\sigma^{(5)}$ 和輔助多項式 $\tau^{(5)}$ 則以下面所示的第(163)式和第(164)式的方式求得。

$$\sigma^{(5)}(x) = \alpha^5 x^4 + \alpha^{10} x^3 + \alpha^{12} x^2 + \alpha^9 x + 1 \quad (163)$$

$$\tau^{(5)}(x) = \alpha^{13} x^3 + x^2 + x + \alpha^6 \quad (164)$$

(請先閱讀背面之注意事項再填寫本頁)

訂

五、發明說明 (84)

而在此 $r=5$ 中，因為 $L=4 > h+t_0=3$ ，所以遞增 r 並且再實施一次 Berlekamp-Massey 演算法計算。

接著，在 $r=6$ ，偏差值 $\Delta_6=\alpha^7$ ，另外，因為 $2L=8$ 、 $r+h=8$ ，所以 $L=4$ 。錯誤位置多項式 $\sigma^{(6)}$ 和輔助多項式 $\tau^{(6)}$ 則以下面所示的第(165)式和第(166)式的方式求得。

$$\sigma^{(6)}(x)=\alpha^6x^3+\alpha^2x^2+\alpha^{10}x+1 \quad (165)$$

$$\tau^{(6)}(x)=\alpha^{13}x^4+x^3+x^2+\alpha^6x \quad (166)$$

而由於此 $\sigma^{(6)}(x)$ 可以因式分解為 $(\alpha^5x+1)(\alpha^7x+1)(\alpha^9x+1)$ ，所以求得根 α^{-5} ， α^{-7} ， α^{-9} ，得知在第 5 成分、第 7 成分以外，第 9 成分上也產生了隨機錯誤。另外，雖然錯誤位置多項式的次數為 3 次，但是由於形式移位暫存器長度 L 為 4，所以得知在第一延伸成分上也產生了錯誤。

接著由於計算步驟數的變數 r 為 6，所以利用下述的第(167)式設定徵候值多項式，再利用第(155)式計算錯誤數值多項式 $\omega(x)$ 。

$$\begin{aligned} S(x) &= S_0 + S_1x + S_2x^2 + S_3x^3 + S_4x^4 + S_5x^5 \\ &= \alpha^{10} + \alpha^{11}x + \alpha^4x^2 + \alpha^{12}x^3 + \alpha^4x^4 + \alpha^{10}x^5 \end{aligned} \quad (167)$$

其結果得到 $\omega(x)=\alpha^{10}+\alpha^3x+\alpha^{14}x^3$ ，做為錯誤數值多項式。藉此，在第 5 成分和第 7 成分上消失的數值 e_5 、 e_7 ，以及在第 9 成分上的隨機錯誤數值 e_9 ，與第九實施例的情況相同，能夠利用如下面所示的第(168)式~第(170)式求得。

$$e_5 = \frac{\omega(\alpha^{-5})}{\sigma'(\alpha^{-5})\alpha^{-5}} = \alpha^{10} \quad (168)$$

(請先閱讀背面之注意事項再填寫本頁)

訂

五、發明說明 (85)

$$e_7 = \frac{\omega(\alpha^{-7})}{\sigma'(\alpha^{-7})\alpha^{-7}} = \alpha^7 \quad (169)$$

$$e_9 = \frac{\omega(\alpha^{-9})}{\sigma'(\alpha^{-9})\alpha^{-9}} = \alpha^{11} \quad (170)$$

藉此，在第十四實施例中，能夠以一次的 Berlekamp-Massey 演算法計算，進行二次延伸 RS 碼的消失/錯誤訂正，另外，藉由比較 Berlekamp-Massey 演算法計算所檢測出的錯誤位置多項式次數和形式移位暫存器長度，亦可得到容易判斷出是否在第一延伸成分中產生錯誤等等的效果。

(請先閱讀背面之注意事項再填寫本頁)

訂

【第十五實施例】

在上述第十三實施例和第十四實施例中，雖然是針對 Berlekamp-Massey 演算法的二次延伸 RS 碼之錯誤訂正方法加以說明，但是亦可以據此實施錯誤訂正裝置。第 26 圖表示在本發明第十五實施例中具有此功能之二階延伸 RS 碼之錯誤訂正裝置的方塊圖。圖中，20 是由收訊字組產生徵候值的徵候值產生裝置。21 是 Berlekamp-Massey 演算法計算裝置，其利用徵候值產生裝置 20 所產生的徵候值實施 Berlekamp-Massey 演算法，並根據形式移位暫存器長度的值，控制是否使用包含第二延伸成分錯誤資訊的徵候值，以便依序計算錯誤位置多項式。

22 是 Chien 搜尋裝置，其對於 Berlekamp-Massey 演算法計算裝置 21 所計算出的錯誤位置多項式進行 Chien 搜尋，以便計算錯誤位置。23 是錯誤數值產生裝

五、發明說明 (86)

置，用來檢查在 Chien 搜尋裝置 22 所計算的錯誤位置之數量是否適當，若適當的話，則利用根據徵候值產生裝置所計算出的徵候值而產生的徵候值多項式，計算錯誤數值。24 是用來儲存輸入之收訊字組之收訊字組儲存裝置。25 則是用來儲存在 Chien 搜尋裝置 22 所計算之錯誤位置，以及在錯誤數值產生裝置 23 中所計算之錯誤數值的計算結果儲存裝置。26 是錯誤訂正裝置，其根據在計算結果儲存裝置 25 中所儲存的錯誤位置和錯誤數值，用來訂正在收訊字組儲存裝置 24 中所儲存之收訊字組的錯誤。

(請先閱讀背面之注意事項再填寫本頁)

接著說明其動作。

在以下的說明中，是採用習知技術中說明的碼長 n 、資訊符號數 k 、最小距離 $d=n-k+1=2t+1$ (t 為正整數) 之二階延伸 RS 碼，並且以收訊字組 $R=(r_{n-2}, r_{n-3}, \dots, r_0, r_1)$ 為所收訊號來做說明。

首先，輸入的收訊字組 R 被送到徵候值產生裝置 20 和收訊字組儲存裝置 24。當徵候值產生裝置 20 接收收訊字組 R ，則產生此收訊字組 R 的徵候值 $S=(S_0, S_1, \dots, S_{2t-2}, S_{2t-1})$ ，再送至 Berlekamp-Massey 演算法計算裝置 21。另外，在收訊字組儲存裝置 24 中則儲存著所輸入原來的收訊字組 R 。

Berlekamp-Massey 演算法計算裝置 21 則利用從徵候值產生裝置 20 所取得的徵候值 S 中除 S_{2t-1} 以外之徵候值 $S_0, S_1, \dots, S_{2t-2}$ ，實施 $2t-1$ 步驟的 Berlekamp-Massey

訂

五、發明說明 (87)

演算法計算。在 $2t-1$ 步驟結束後，若形式移位暫存器長度 L 在 $t-1$ 以下，則輸出此時的錯誤位置多項式，並結束 Berlekamp-Massey 演算法計算。另外，若形式移位暫存器長度 L 在 t 以上，則利用徵候值 S_{2t-1} ，再進行一次 Berlekamp-Massey 演算法計算，輸出此時的錯誤位置多項式並結束 Berlekamp-Massey 演算法計算。

藉此在 Berlekamp-Massey 演算法計算裝置 21 中所計算的錯誤位置多項式，則被送到 Chien 搜尋裝置 22。在 Chien 搜尋裝置 22 中則對此錯誤位置多項式進行 Chien 搜尋，以計算出此錯誤位置多項式的根(錯誤位置)。在 Chien 搜尋裝置 22 所算出的錯誤位置多項式之根，被保存在計算結果儲存裝置 25 中，同時被送到錯誤數值產生裝置 23。錯誤數值產生裝置 23 則是在 Chien 搜尋中所得的根數量和錯誤位置多項式的次數一致時，計算出在錯誤位置上所產生的錯誤大小，並將其儲存在計算結果儲存裝置 25 中。

錯誤訂正裝置 26 在能夠訂正時，根據在計算結果儲存裝置 25 中儲存的內容，以對收訊字組儲存裝置 24 中儲存之收訊字組 R 中的錯誤位置，減去其錯誤大小的方式，進行錯誤的訂正，並輸出訂正後的結果。另外，當無法進行訂正時，則將錯誤檢測旗標連同原來的收訊字組輸出。

藉此，在第十五實施例中，能夠以一次 Berlekamp-Massey 演算法計算，進行二階延伸 RS 碼的錯誤訂正，

(請先閱讀背面之注意事項再填寫本頁)

訂

五、發明說明 (88)

另外，只需要略為修改習知的二階延伸 RS 碼錯誤訂正裝置，便可以構成二階延伸 RS 碼錯誤訂正裝置，所以可以得到減少成本的效果。

(請先閱讀背面之注意事項再填寫本頁)

訂

【第十六實施例】

在上述第十五實施例中，雖然是針對以 Berlekamp-Massey 演算法來進行二階延伸 RS 碼錯誤訂正的二階延伸 RS 碼錯誤訂正裝置進行說明，但是也可以適用於用來進行二階延伸 RS 碼消失/錯誤訂正的二階延伸 RS 碼錯誤訂正裝置。第 27 圖表示本發明第十六實施例中具有此功能之二階延伸 RS 碼之錯誤訂正裝置的方塊圖。圖中，27 是消失位置係數產生裝置，其利用伴隨在輸入收訊字組之消失旗標，產生消失位置係數。28 是消失位置多項式產生裝置，由消失位置係數產生裝置 27 所產生的消失位置係數，產生消失位置多項式。

而其他部分與第十五實施例中第 26 圖附上相同符號者，則為對應的部分，但是 Berlekamp-Massey 演算法計算裝置 21 則與第十五實施例之第 26 圖中附上相同標記者不同，其差異點在於將消失位置多項式產生裝置 28 中所計算的消失位置多項式，設定為初始值，以進行 Berlekamp-Massey 演算法，而藉由形式移位暫存器長度的值，控制是否使用包含第二延伸成分錯誤資訊的徵候值，以依序計算錯誤位置多項式。

接著說明其動作。

五、發明說明 (89)

在以下的說明中，是採用習知技術中說明的碼長 n 、資訊符號數 k 、最小距離 $d=n-k+1=2t+1$ (t 為正整數) 之二階延伸 RS 碼進行說明。另外，係以收訊字組 $R=(r_{n-2}, r_{n-3}, \dots, r_0, r_1)$ 為所收訊號，並在內部碼中產生 h 個消失錯誤的情況來做說明。另外，設 $t_0=[(2t-h-1)/2]$ 。

首先，輸入的收訊字組 R 被送到徵候值產生裝置 20，消失位置係數產生裝置 27 和收訊字組儲存裝置 24。徵候值產生裝置 20 利用所取得的收訊字組 R ，產生徵候值 $S=(S_0, S_1, \dots, S_{2t-2}, S_{2t-1})$ ，並送到 Berlekamp-Massey 演算法計算裝置 21。另外，消失位置係數產生裝置 27 根據伴隨在收訊字組 R 的消失旗標，產生消失位置係數，並送到消失位置多項式產生裝置 28。消失位置多項式產生裝置 28 則利用此消失位置係數產生消失位置多項式，送出到 Berlekamp-Massey 演算法計算裝置 21。而在收訊字組儲存裝置 24 中，則是儲存著所輸入原來的收訊字組 R 。

Berlekamp-Massey 演算法計算裝置 21 則利用從徵候值產生裝置 20 所取得的徵候值 S 中除 S_{2t-1} 以外之徵候值 $S_0, S_1, \dots, S_{2t-2}$ 實施 $2t-1$ 步驟的 Berlekamp-Massey 演算法計算。在 $2t-1$ 步驟結束後，若形式移位暫存器長度 L 在 $h+t_0$ 以下，則輸出此時的錯誤位置多項式，並結束 Berlekamp-Massey 演算法計算。另外，若形式移位暫存器長度 L 在 $h+t_0+1$ 以上，則利用徵候值 S_{2t-1} ，再返回進行一次 Berlekamp-Massey 演算法計算，輸出此時的錯

(請先閱讀背面之注意事項再填寫本頁)

訂

五、發明說明 (90)

誤位置多項式並結束 Berlekamp-Massey 演算法計算。

藉此在 Berlekamp-Massey 演算法計算裝置 21 中所計算的錯誤位置多項式，則被送到 Chien 搜尋裝置 22。在 Chien 搜尋裝置 22 中則對此錯誤位置多項式進行 Chien 搜尋，以計算其根(消失/錯誤位置)。在 Chien 搜尋裝置 22 所算出的錯誤位置多項式之根，被保存在計算結果儲存裝置 25 中，同時被送到錯誤數值產生裝置 23。錯誤數值產生裝置 23 則是在 Chien 搜尋中所得的根數量和錯誤位置多項式的次數一致時，計算出在錯誤/消失位置上所產生的錯誤大小，並將其儲存在計算結果儲存裝置 25 中。

錯誤訂正裝置 26 在能夠訂正時，根據在計算結果儲存裝置 25 中儲存的內容，以對收訊字組儲存裝置 24 中儲存之收訊字組 R 中的錯誤位置，減去其錯誤大小的方式，進行錯誤的訂正，並輸出訂正後的結果。另外，當無法進行訂正時，則將錯誤檢測旗標連同原來的收訊字組 R 輸出。

藉此，在第十六實施例中，能夠以一次 Berlekamp-Massey 演算法計算，進行二階延伸 RS 碼的錯誤訂正，另外，只需要略為修改習知的二階延伸 RS 碼錯誤訂正裝置，便可以構成二階延伸 RS 碼錯誤訂正裝置，所以可以得到減少成本的效果。

【第十七實施例】

(請先閱讀背面之注意事項再填寫本頁)

訂

五、發明說明 (91)

在上述第十三~十六實施例中，雖然是針對以 Berlekamp-Massey 演算法進行二階延伸 RS 碼之錯誤訂正情況進行說明，但是也可以對於訂正能力較小的二階延伸 RS 碼，以預先估算錯誤個數的方式而使用歐氏演算法的情況。以下在本發明第十七實施例中，係採用碼長 17、資訊符號數 11、最小距離 7 的高氏場 GF(16)上之二階延伸 RS 碼來做說明。而原始多項式和配類陣列則與第十三實施例之情況相同。另外在以下的說明中，係以收訊字組 $R=(r_{15}, r_{14}, \dots, r_0, r_{-1})$ 為所收訊號的情況來說明。

首先，由輸入的收訊字組 R ，計算徵候值 $S_0, S_1, \dots, S_4, S_5$ 。藉此，如以下第(171)式所示之方式，構成陣列 A ，再計算其行列式 \det 。

$$A = \begin{bmatrix} S_0 & S_1 & S_2 \\ S_1 & S_2 & S_3 \\ S_2 & S_3 & S_4 \end{bmatrix} \quad (171)$$

當在內部碼和第一延伸成分上產生了 3 個錯誤時，則行列式 \det 不為 0。此處假設在 $i, j, k (0 \leq i, j, k \leq 14)$ 上產生了錯誤 e_i, e_j, e_k 。此時徵候值則成為下面的第(172)式。

$$S_m = e_i \alpha^{im} + e_j \alpha^{jm} + e_k \alpha^{km} \quad (m = 0, 1, 2, \dots, 4, 5) \quad (172)$$

因此，上述第(171)式所示之陣列 A ，即轉換為下面第(173)式。

(請先閱讀背面之注意事項再填寫本頁)

訂

五、發明說明 (92)

$$A = \begin{bmatrix} 1 & 1 & 1 \\ \alpha^i & \alpha^j & \alpha^k \\ \alpha^{2i} & \alpha^{2j} & \alpha^{2k} \end{bmatrix} \begin{bmatrix} e_i & 0 & 0 \\ 0 & e_j & 0 \\ 0 & 0 & e_k \end{bmatrix} \begin{bmatrix} 1 & \alpha^i & \alpha^{2i} \\ 1 & \alpha^j & \alpha^{2j} \\ 1 & \alpha^k & \alpha^{2k} \end{bmatrix} \quad (173)$$

因此在此第(173)式右邊的陣列中，中間者為對角化陣列的正則 (regular) 陣列，左右者則為所謂 Vandermonde 陣列的正則陣列，所以陣列 A 的行列式 \det 不為 0。

另外如第一延伸成分和 $i, j (0 \leq i, j \leq 14)$ 上是產生錯誤 e_{-i} 和 e_i, e_j 。此時徵候值即成為下面第(174)式和第(175)式。

$$S_0 = e_i + e_j + e_{-i} \quad (174)$$

$$S_m = e_i \alpha^{im} + e_j \alpha^{jm} \quad (m=1, 2, \dots, 4, 5) \quad (175)$$

因此，第(171)式所示之陣列 A 即轉換成下面第(176)式的形態。

$$A = \begin{bmatrix} 1 & 1 & 1 \\ \alpha^i & \alpha^j & 0 \\ \alpha^{2i} & \alpha^{2j} & 0 \end{bmatrix} \begin{bmatrix} e_i & 0 & 0 \\ 0 & e_j & 0 \\ 0 & 0 & e_k \end{bmatrix} \begin{bmatrix} 1 & \alpha^i & \alpha^{2i} \\ 1 & \alpha^j & \alpha^{2j} \\ 1 & 0 & 0 \end{bmatrix} \quad (176)$$

由於在第(176)式右邊的陣列全部是正則陣列，所以陣列 A 的行列式 \det 不為 0。另一方面，若內部碼和第一延伸成分上產生了兩個以下的錯誤時，則行列式 \det 為 0。例如在第(173)式中，當 $e_k=0$ 時，由於在第(173)式中間的陣列不是正則陣列，所以陣列 A 也不是正則陣列。

利用以上所述，使用陣列 A 的行列式 \det ，能夠估算出錯誤的個數。第 28 圖表示在第十七實施例中，二階

(請先閱讀背面之注意事項再填寫本頁)

訂

五、發明說明 (93)

延伸 RS 碼之錯誤訂正方式處理程序的流程圖，係以上述行列式估算錯誤的個數，再藉由歐氏演算法進行二階延伸 RS 碼的錯誤訂正。圖中，ST41 是以收訊字組計算出其徵候值的徵候值產生步驟，ST42 是對使用徵候值產生步驟 ST41 來計算出徵候值而構成的陣列，計算行列式之步驟。另外，ST43 則是判斷是否在行列式計算步驟 ST42 中所計算出的行列式為 0 的判斷步驟。

ST44a 是針對在判斷步驟 ST43 中判斷出行列式不為 0 的情況下，另外 ST44b 則是針對在判斷步驟 ST43 中判斷出行列式為 0 的情況下，設定用來計算錯誤位置多項式和錯誤數值多項式的初始值和終止條件，並且藉由歐氏演算法，計算錯誤位置多項式和錯誤數值多項式之歐氏演算法計算步驟。在此歐氏演算法計算步驟 ST44a 和 ST44b 中的歐氏演算法計算程序，則與第 31 圖所示之習知情況相同。而上述 ST44a 和 ST44b 則稱為第一歐氏演算法計算步驟和第二歐氏演算法計算步驟，以區分此兩者。

另外，ST45a 是將第一歐氏演算法計算步驟 ST44a 所計算出的錯誤位置多項式次數與錯誤數值多項式次數進行比較的第一比較步驟，ST45b 是將第二歐氏演算法計算步驟 ST44b 所計算出的錯誤位置多項式次數與錯誤數值多項式次數進行比較的第二比較步驟。

ST46 是在第一比較步驟 ST45a 中判斷出錯誤數值多項式次數不高於錯誤位置多項式次數的情況，或是在

(請先閱讀背面之注意事項再填寫本頁)

訂

五、發明說明 (94)

第二比較步驟 ST45b 中判斷出錯誤數值多項式次數未達錯誤位置多項式次數的情況，對此錯誤位置多項式和錯誤數值多項式進行 Chien 搜尋，以便計算出錯誤位置多項式的根(錯誤位置)和錯誤數值的 Chien 搜尋步驟。ST47 則是判斷在 Chien 搜尋步驟 ST46 所得之錯誤位置多項式的根(錯誤位置)數量是否適當的檢定步驟。ST48 則是當檢定步驟 ST47 所判斷結果顯示根數量適當時，根據 Chien 搜尋步驟 ST46 所得到的錯誤位置和錯誤數值，來訂正輸入之收訊字組 R 中錯誤的錯誤訂正步驟。

(請先閱讀背面之注意事項再填寫本頁)

接著說明其動作。

首先在徵候值產生步驟 ST41 中，由輸入之收訊字組 R，計算徵候值 $S_0, S_1, \dots, S_4, S_5$ 。接著在步驟 ST42 中，若在徵候值產生步驟 ST41 中所產生的徵候值全部為 0 時，則推測沒有錯誤。另外，若不是所有徵候值為 0 的情況，則計算上述陣列 A 的行列式 det。

接著在判斷步驟 ST43 中判斷是否其行列式 det 為 0。當行列式 det 不為 0 時，即推測在碼內產生了 3 個符號的錯誤，而進入第一歐氏演算法計算步驟 ST44a，利用徵候值 $S_0, S_1, \dots, S_4, S_5$ ，進行歐氏演算法。亦即，設定包含第一延伸成分錯誤資訊的徵候值 S_0 為常數項，以構成徵候值多項式，再以下面第(177)式和第(178)式的方式，設定歐氏演算法的初始值。

$$R_{-1}(x) = x^6 \quad (177)$$

$$R_0(x) = S_0 + S_1x + S_2x^2 + S_3x^3 + S_4x^4 + S_5x^5 \quad (178)$$

五、發明說明 (95)

如第 31 圖所示，遞增 i 並依序構成 R_i ，當此 R_i 滿足下面所示第(179)式時，則結束第一歐氏演算法計算步驟 ST44a。

$$\deg R_i < 3 \quad (179)$$

在第一歐氏演算法計算步驟 ST44a 結束後，則在第一比較步驟 ST45a 中比較所得之錯誤位置多項式 $\sigma(x)$ 和錯誤數值多項式 $\omega(x)$ 次數，以確定其是否滿足下面第(180)式所示之不等式。

$$\deg \sigma \geq \deg \omega \quad (180)$$

當結果顯示滿足上述第(180)式所示之條件時，則進行 Chien 搜尋步驟 ST46 的處理。而當第(180)式的等號成立時，即推測在第一延伸成分中產生了錯誤。在 Chien 搜尋步驟 ST46 中，對上述的錯誤位置多項式 $\sigma(x)$ 和錯誤數值多項式 $\omega(x)$ 進行 Chien 搜尋，以便計算錯誤位置多項式 $\sigma(x)$ 的根(錯誤位置)和錯誤數值。而錯誤數值能夠利用第(123)式加以計算。另一方面，當不滿足此條件時，則不能夠進行錯誤的訂正，僅止於錯誤的檢測。

另外，如果在判斷步驟 ST43 中的判斷結果表示上述行列式為 0 時，則推測在內部碼中產生了兩個符號以下的錯誤，並進行第二歐氏演算法計算步驟 ST44b，利用微候值 S_1, S_2, S_3, S_4 實施歐氏演算法。而此時歐氏演算法的初始值，則可利用以下第(181)式和第(182)式的方式加以設定。

$$R_{-1}(x) = x^4 \quad (181)$$

(請先閱讀背面之注意事項再填寫本頁)

訂

五、發明說明 (96)

$$R_0(x) = S_1 + S_2x + S_3x^2 + S_4x^3 \quad (182)$$

如第 31 圖所示，遞增 i 並且依序構成 R_i ，當此 R_i 值滿足下面所示之第(183)式時，則結束此第二歐氏演算法計算步驟 ST44b。

$$\deg R_i < 2 \quad (183)$$

在結束此第二歐氏演算法計算步驟 ST44b 之後，在第二比較步驟 ST45b 中比較所得之錯誤位置多項式 $\sigma(x)$ 和錯誤數值多項式 $\omega(x)$ 之次數，以確定其是否滿足下面第(184)式所示之不等式。

$$\deg \sigma > \deg \omega \quad (184)$$

當其結果顯示上述第(184)式所示條件滿足時，則進行 Chien 搜尋步驟 ST46。在 Chien 搜尋步驟 ST46 中，對於上述錯誤位置多項式 $\sigma(x)$ 和錯誤數值多項式 $\omega(x)$ 進行 Chien 搜尋，以計算出錯誤位置多項式 $\sigma(x)$ 的根(錯誤位置)以及錯誤數值。另一方面，當不滿足此條件時，則無法進行錯誤訂正，而只能進行錯誤的檢測。

當在 Chien 搜尋步驟 ST46 中算出錯誤位置和錯誤數值之後，則在檢定步驟 ST47 中，判斷是否所獲得適當的根個數，亦即判斷是否錯誤位置的個數等於錯誤位置多項式 $\sigma(x)$ 的次數。當相等時，則在錯誤位置上減去錯誤數值，以便訂正收訊字組的錯誤，並輸出訂正結果。另外，當錯誤位置的個數和錯誤位置多項式 $\sigma(x)$ 的次數不相等時，則無法進行錯誤的訂正，而將錯誤檢出旗標連同原來之收訊字組輸出。

(請先閱讀背面之注意事項再填寫本頁)

訂

(請先閱讀背面之注意事項再填寫本頁)

五、發明說明 (97)

藉此，在第十，對於訂正能力較小的二階延伸 RS 碼而言，候值以預先估算出錯誤的個數，並且能夠以演算法計算來計算出錯誤位置多項式和錯誤數值。此時，由於錯誤數值多項式可由歐氏演算法產物而獲得，所以不需要為了獲得錯誤數值進行特別的計算。

【第十八實施例】

在上述第十七實雖然是針對利用歐氏演算法的二階延伸 RS 碼方法加以說明，但是亦可據此實現錯誤訂正裝置圖表示本發明第十八實施例中具有此功能之二RS 碼錯誤訂正裝置的方塊圖。圖中，30 是由收生徵候值的徵候值產生裝置。31 是錯誤個數估其構成在徵候值產生裝置 30 所計算的徵候值陣列式，以估算其錯誤個數。32 是歐氏演算置，其對應於錯誤個數估算裝置 31 所估算出的，設定用來計算錯誤位置多項式和錯誤數值多項式以及終止條件，再利用歐氏演算法，計算多項式和錯誤數值多項式。

33 是 Chien 搜尋裝置對於歐氏演算法計算裝置 32 所計算出的錯誤位置和錯誤數值多項式進行 Chien 搜尋，以便計算和錯誤數值。34 是用來儲存輸入之收訊字組之儲存裝置。35 則是用來

五、發明說明 (98)

儲存在 Chien 搜尋裝置 22 所計算之錯誤位置和錯誤數值的計算結果儲存裝置。36 是錯誤訂正裝置，其根據在計算結果儲存裝置 35 中所儲存的錯誤位置和錯誤數值，用來訂正在收訊字組儲存裝置 34 中所儲存之收訊字組的錯誤。

接著說明其動作。

在以下的說明中，是採用在高氏場 GF(16)上，碼長 17、資訊符號數 11、最小距離 7 之二階延伸 RS 碼。另外，原始方程式和配類陣列則與第十七實施例相同。

首先，輸入的收訊字組被送到徵候值產生裝置 30 和收訊字組儲存裝置 34。當徵候值產生裝置 30 接收收訊字組之後，則產生此收訊字組的徵候值 $S = (S_0, S_1, \dots, S_4, S_5)$ ，再送至錯誤個數估算裝置 31。另外，在收訊字組儲存裝置 34 中則儲存著所輸入原來的收訊字組。

錯誤個數估算裝置 31 則利用從徵候值產生裝置 30 所取得的徵候值 S 中，除了受第二延伸成分影響的徵候值 S_5 以外者，以便計算由第(171)式所定義之陣列 A 的行列式 $\det A$ 。當其值為 0，則推測產生了 2 個符號以下的錯誤，若不為 0 則推測產生了 3 個符號的錯誤。

當推測產生了 3 個符號錯誤時，則在歐氏演算法計算裝置 32 中，分別利用第(177)式和第(178)式設定初始值，並利用第(179)式設定終止條件，以便進行歐氏演算法計算，計算出錯誤位置多項式 $\sigma(x)$ 和錯誤數值多項式 $\omega(x)$ 。接著比較錯誤位置多項式 $\sigma(x)$ 和錯誤數值多項式

(請先閱讀背面之注意事項再填寫本頁)

訂

五、發明說明 (99)

$\omega(x)$ 的次數，若第(180)式所給定的條件不成立時，則僅進行錯誤的檢測，並且輸出原來的收訊字組。另一方面，若第(180)式的條件成立時，則在 Chien 搜尋裝置 33 中對錯誤位置多項式 $\sigma(x)$ 和錯誤數值多項式 $\omega(x)$ 進行 Chien 搜尋，計算出錯誤位置和錯誤數值後，儲存在計算結果儲存裝置 35 中。

另外，當推測產生了 2 個符號以下的錯誤時，則在歐氏演算法計算裝置 32 中，分別利用第(181)式和第(182)式設定初始值，並利用第(183)式設定終止條件，以便進行歐氏演算法計算，計算出錯誤位置多項式 $\sigma(x)$ 和錯誤數值多項式 $\omega(x)$ 。接著比較錯誤位置多項式 $\sigma(x)$ 和錯誤數值多項式 $\omega(x)$ 的次數，若第(184)式所給定的條件不成立時，則僅進行錯誤的檢測，並且輸出原來的收訊字組。另一方面，若第(184)式的條件成立時，則在 Chien 搜尋裝置 33 中對錯誤位置多項式 $\sigma(x)$ 和錯誤數值多項式 $\omega(x)$ 進行 Chien 搜尋，計算出錯誤位置和錯誤數值後，儲存在計算結果儲存裝置 35 中。

當在 Chien 搜尋裝置 33 中所檢測的錯誤個數和錯誤位置多項式的次數相等時，則在錯誤訂正裝置 36 中，對於收訊字組儲存裝置 34 中所儲存的收訊字組之各錯誤位置上，減去其錯誤數值，藉此進行錯誤的訂正，並且輸出訂正的結果。另一方面，當所檢出的錯誤個數與錯誤位置多項式的次數不一致時，則僅進行錯誤的檢測，並且輸出原來的收訊字組。

(請先閱讀背面之注意事項再填寫本頁)

訂

五、發明說明(00)

藉此，對於訂正能力較小的二階延伸 RS 碼而言，係採用徵候值以預先估算出錯誤的個數，並且能夠以一次的歐氏演算法計算來計算出錯誤位置多項式和錯誤數值多項式。此時，由於錯誤數值多項式可由歐氏演算法計算裝置的計算副產物而獲得，所以不需要為了獲得錯誤數值多項式而進行特別的計算。

由於在本發明中的延伸 RS 碼錯誤訂正解碼裝置，錯誤個數估算裝置係根據徵候值產生裝置所計算出的收訊字組徵候值，估算錯誤個數，接著藉由採用此錯誤個數所設定的初始值和終止條件、由歐氏演算法計算裝置所計算出的錯誤位置多項式和錯誤數值多項式，在 Chien 搜尋裝置中計算出錯誤位置和錯誤數值，以便由錯誤訂正裝置據此進行收訊字組的錯誤訂正，所以分別藉由各一次的歐氏演算法計算裝置之操作和 Chien 搜尋裝置之操作，便能夠解碼延伸 RS 碼，而能夠高速地進行解碼操作。

另外，利用徵候值修正裝置中高氏場上的加法/乘法演算裝置，由 Chien 搜尋裝置所算出之錯誤位置和錯誤數值來修正收訊字組的徵候值，所以不僅對於延伸成分以外的錯誤數值，同時對於延伸成分的錯誤數值，也能夠以簡單的電路結構，輕易地計算。

另外，由於係根據高氏場上乘法裝置、加法裝置和儲存裝置所構成的錯誤個數估算裝置，判斷出延伸 RS 碼的最大錯誤訂正個數低於 $[(d-1)/2]$ 個或是等於 $[(d-1)/2]$

(請先閱讀背面之注意事項再填寫本頁)

訂

五、發明說明(01)

個的判斷結果，而改變歐氏演算法計算裝置在上述錯誤個數估算裝置中對於歐氏演算法計算操作的初始值和終止條件，所以能夠減少解碼時所需要的步驟。

另外，由於在錯誤個數估算裝置中判斷低於、等於、高於延伸 RS 碼的最大錯誤訂正個數時，在高於最大錯誤訂正個數的情況下，歐氏演算法計算裝置中止錯誤位置多項式和錯誤數值多項式之計算，而 Chien 搜尋裝置則中止錯誤位置和錯誤數值之計算，所以能夠減少解碼時所需要的步驟。

上述錯誤訂正裝置在上述錯誤個數估算裝置判斷出錯誤個數高於 $[(d-1)/2]$ 個時，輸出儲存在上述收訊字組儲存裝置中之原來收訊字組。

本發明中的一階延伸李德-所羅門碼之錯誤訂正裝置，由於徵候值多項式產生裝置，係將收訊字組徵候值中包含延伸成分錯誤資訊者設為常數項之方式，產生徵候值多項式，而歐氏演算法計算裝置則將其設為初始值，以求得錯誤位置多項式和錯誤數值多項式，再使用 Chien 搜尋裝置據此所計算的錯誤位置和錯誤數值，進行錯誤的訂正，所以因為以一次歐氏演算法計算裝置中的操作，而得到錯誤位置多項式和錯誤數值多項式，故能夠高速地進行錯誤訂正操作的處理。

另外，由於利用消失位置係數產生裝置和消失位置多項式產生裝置以伴隨於收訊字組的消失旗標所產生的消失位置多項式，以及徵候值多項式產生裝置所產生之

(請先閱讀背面之注意事項再填寫本頁)

訂

五、發明說明 (102)

徵候值多項式，修正徵候值多項式產生裝置可產生修正徵候值多項式，歐氏演算法計算裝置將其設為初始值以求得錯誤消失位置多項式和錯誤消失數值多項式，Chien搜尋裝置則據此計算出錯誤位置和錯誤的大小，以便進行錯誤訂正，所以能夠有效率地進行一階延伸 RS 碼的消失/錯誤訂正。

本發明中的二階延伸李德-所羅門碼之錯誤訂正裝置，由於 Berlekamp-Massey 演算法計算裝置係採用收訊字組的徵候值，依序計算錯誤位置多項式，而當 Chien 搜尋裝置據此所計算之錯誤位置數量適當時，錯誤數值產生裝置則利用以收訊字組徵候值所得之徵候值多項式，求得錯誤數值，以便據此進行錯誤的訂正，所以能夠高速地進行訂正的操作。

另外，由於將消失位置係數產生裝置和消失位置多項式產生裝置藉由伴隨於收訊字組之消失旗標所產生的消失位置多項式，設為初始值，Berlekamp-Massey 演算法計算裝置則依序產生錯誤位置多項式，所以能夠有效率地進行二階延伸 RS 碼的消失/錯誤訂正。

圖式之簡單說明：

第 1 圖表示本發明第一實施例中，延伸 RS 碼之錯誤訂正解碼方法之處理程序的流程圖

第 2 圖表示上述第一實施例中，徵候值產生步驟之處理程序的詳細流程圖。

(請先閱讀背面之注意事項再填寫本頁)

訂

五、發明說明 (103)

第 3 圖表示上述第一實施例中，錯誤個數估算步驟之處理程序的詳細流程圖。

第 4 圖表示上述第一實施例中，歐氏演算法計算步驟之處理程序的詳細流程圖。

第 5 圖表示上述第一實施例中，Chien 搜尋步驟之處理程序的詳細流程圖。

第 6 圖表示上述第一實施例中，錯誤訂正步驟之處理程序的詳細流程圖。

第 7 圖表示本發明第二實施例之延伸 RS 碼錯誤訂正解碼方法中，Chien 搜尋步驟之處理程序的詳細流程圖。

第 8 圖表示本發明第三實施例中，延伸 RS 碼之錯誤訂正解碼方法之處理程序的流程圖。

第 9 圖表示上述第三實施例中，錯誤個數估算步驟之處理程序的詳細流程圖。

第 10 圖表示本發明第四實施例中，延伸 RS 碼之錯誤訂正解碼方法之處理程序的流程圖。

第 11 圖表示上述第四實施例中，錯誤個數估算步驟之處理程序的詳細流程圖。

第 12 圖表示本發明第五實施例中，延伸 RS 碼之錯誤訂正解碼裝置的方塊圖。

第 13 圖表示本發明第六實施例中，延伸 RS 碼之錯誤訂正解碼裝置的方塊圖。

第 14 圖表示本發明第七實施例中，延伸 RS 碼之錯

(請先閱讀背面之注意事項再填寫本頁)

訂

五、發明說明(104)

誤訂正解碼裝置的方塊圖。

第 15 圖表示本發明第八實施例中，延伸 RS 碼之錯誤訂正解碼裝置的方塊圖。

第 16 圖表示本發明第九實施例中，一階延伸 RS 碼之錯誤訂正方法之處理程序的流程圖。

第 17 圖表示上述第九實施例中，歐氏演算法計算步驟之處理程序的詳細流程圖。

第 18 圖表示本發明第十實施例中，一階延伸 RS 碼之錯誤訂正方法之處理程序的流程圖。

第 19 圖表示上述第十實施例中，歐氏演算法計算步驟之處理程序的詳細流程圖。

第 20 圖表示本發明第十一實施例中，一階延伸 RS 碼之錯誤訂正解碼裝置的方塊圖。

第 21 圖表示本發明第十二實施例中，一階延伸 RS 碼之錯誤訂正解碼裝置的方塊圖。

第 22 圖表示本發明第十三實施例中，二階延伸 RS 碼之錯誤訂正方法之處理程序的流程圖。

第 23 圖表示上述第十三實施例中，Berlekamp-Massey 演算法計算步驟之處理程序的詳細流程圖。

第 24 圖表示本發明第十四實施例中，二階延伸 RS 碼之錯誤訂正方法之處理程序的流程圖。

第 25 圖表示上述第十四實施例中，Berlekamp-Massey 演算法計算步驟之處理程序的詳細流程圖。

第 26 圖表示本發明第十五實施例中，二階延伸 RS

(請先閱讀背面之注意事項再填寫本頁)

訂

五、發明說明(105)

碼之錯誤訂正解碼裝置的方塊圖。

第 27 圖表示本發明第十六實施例中，二階延伸 RS 碼之錯誤訂正解碼裝置的方塊圖。

第 28 圖表示本發明第十七實施例中，二階延伸 RS 碼之錯誤訂正方法之處理程序的流程圖。

第 29 圖表示本發明第十八實施例中，二階延伸 RS 碼之錯誤訂正解碼裝置的方塊圖。

第 30 圖表示習知一階延伸 RS 碼之錯誤訂正方法之處理程序的流程圖。

第 31 圖表示上述一階延伸 RS 碼之錯誤訂正方法中，歐氏演算法計算步驟之處理程序的詳細流程圖。

第 32 圖表示習知二階延伸 RS 碼之錯誤訂正方法之處理程序的流程圖。

第 33 圖表示上述二階延伸 RS 碼之錯誤訂正方法中，Berlekamp-Massey 演算法計算步驟之處理程序的詳細流程圖。

符號說明：

ST11、ST21、ST31、ST41~徵候值產生步驟；ST12~錯誤個數估算步驟；ST13、ST23~歐氏演算法計算步驟；ST13a、ST44a~第一歐氏演算法計算步驟(歐氏演算法計算步驟)；ST13b、ST44b~第二歐氏演算法計算步驟(歐氏演算法計算步驟)；ST14、ST25、ST33、ST46~Chien 搜尋步驟；ST15、ST26、ST35、ST48~錯誤訂正步驟；

(請先閱讀背面之注意事項再填寫本頁)

訂

五、發明說明 (106)

ST15a~錯誤訂正操作步驟(錯誤訂正步驟)；ST15b~錯誤檢出操作步驟(錯誤訂正步驟)；ST22~徵候值多項式產生步驟；ST27~消失位置係數產生步驟；ST28、ST36~消失位置多項式產生步驟；ST29~修正徵候值多項式產生步驟；ST32~Berlekamp-Massey 演算法計算步驟；1、10、20、30~徵候值產生裝置；2、9、31~錯誤個數估算裝置；3、12、32~歐氏演算法計算裝置；4、13、22、33~Chien搜尋裝置；5、14、24、34~收訊字組儲存裝置；7、16、26、36~錯誤訂正裝置；8~徵候值修正裝置；11~徵候值多項式產生裝置；17、27~消失位置係數產生裝置；18、28~消失位置多項式產生裝置；19~修正徵候值多項式產生裝置；21~Berlekamp-Massey 演算法計算裝置；23~錯誤數值產生裝置。

(請先閱讀背面之注意事項再填寫本頁)

訂

六、申請專利範圍

1.一種延伸李德-所羅門碼之錯誤訂正解碼裝置，其包括：

徵候值產生裝置，用以根據所輸入之收訊字組，計算出其徵候值；

收訊字組儲存裝置，用以儲存上述收訊字組；

錯誤個數估算裝置，根據上述徵候值產生裝置所計算出的徵候值，估算錯誤個數；

歐氏演算法計算裝置，其對應於上述錯誤個數估算裝置所估算出之錯誤個數，設定用以計算錯誤位置多項式和錯誤數值多項式之初始值和終止條件，並且藉由歐氏演算法，計算錯誤位置多項式和錯誤數值多項式；

Chien 搜尋裝置，根據上述歐氏演算法計算裝置所計算出之錯誤位置多項式和錯誤數值多項式，計算錯誤位置和錯誤數值；以及

錯誤訂正裝置，根據上述 Chien 搜尋裝置的計算結果，對儲存於上述收訊字組儲存裝置中之收訊字組，訂正其錯誤。

2.如申請專利範圍第 1 項所述之延伸李德-所羅門碼之錯誤訂正解碼裝置，係藉由具有高氏場上加法裝置和乘法裝置之演算裝置所構成，其中上述 Chien 搜尋裝置設置有徵候值修正裝置，係藉由基於上述歐氏演算法計算裝置所計算出的錯誤位置多項式和錯誤數值多項式，所算出之錯誤位置和錯誤數值，以修正收訊字組徵候值的方式，計算出延伸成分中的錯誤數值。

(請先閱讀背面之注意事項再填寫本頁)

訂

六、申請專利範圍

3.如申請專利範圍第1項所述之延伸李德-所羅門碼之錯誤訂正解碼裝置，上述錯誤個數估算裝置係由高氏場上乘法裝置、加法裝置和儲存裝置所構成，對於最小距離為 d 之延伸李德-所羅門碼，係根據在上述徵候值產生裝置中所計算之收訊字組的徵候值，判斷是否有錯誤，在存在錯誤時，則判斷其個數低於 $[(d-1)/2]$ 個或是等於 $[(d-1)/2]$ 個；

上述歐氏演算法計算裝置在上述錯誤個數估算裝置判斷出存在錯誤時，根據錯誤個數是低於 $[(d-1)/2]$ 個或是等於 $[(d-1)/2]$ 個之情況，進行用以計算錯誤位置多項式和錯誤數值多項式之初始值和終止條件的設定，藉由歐氏演算法，計算錯誤位置多項式和錯誤數值多項式。

4.如申請專利範圍第1項所述之延伸李德-所羅門碼之錯誤訂正解碼裝置，上述錯誤個數估算裝置對於最小距離為 d 之延伸李德-所羅門碼，係根據在上述徵候值產生裝置中所計算之收訊字組的徵候值，判斷是否有錯誤，在存在錯誤時，則判斷其個數低於 $[(d-1)/2]$ 個、等於 $[(d-1)/2]$ 個或高於 $[(d-1)/2]$ 個；

上述歐氏演算法計算裝置在上述錯誤個數估算裝置判斷出錯誤個數高於 $[(d-1)/2]$ 個時，中止藉由歐氏演算法對錯誤位置多項式和錯誤數值多項式之計算；

上述 Chien 搜尋裝置在上述錯誤個數估算裝置判斷出錯誤個數高於 $[(d-1)/2]$ 個時，中止藉由 Chien 搜尋對錯誤位置和錯誤數值之計算；

(請先閱讀背面之注意事項再填寫本頁)

訂

六、申請專利範圍

上述錯誤訂正裝置在上述錯誤個數估算裝置判斷出錯誤個數高於 $[(d-1)/2]$ 個時，輸出儲存在上述收訊字組儲存裝置中之原來收訊字組。

5.一種一階延伸李德-所羅門碼之錯誤訂正裝置，其包括：

徵候值產生裝置，用以根據所輸入之收訊字組，計算出其徵候值；

收訊字組儲存裝置，用以儲存上述收訊字組；

徵候值多項式產生裝置，係對上述徵候值產生裝置所計算出者之中包含延伸成分錯誤資訊的徵候值設為常數項之方式，產生徵候值多項式；

歐氏演算法計算裝置，用以將上述徵候值多項式產生裝置所產生之徵候值多項式，設為初始值，並且藉由歐氏演算法，計算錯誤位置多項式和錯誤數值多項式；

Chien 搜尋裝置，根據上述歐氏演算法計算裝置所計算出之錯誤位置多項式和錯誤數值多項式，計算錯誤位置和錯誤數值；以及

錯誤訂正裝置，根據上述 Chien 搜尋裝置所計算之錯誤位置和錯誤數值，用以訂正儲存於上述收訊字組儲存裝置中之收訊字組的錯誤。

6.如申請專利範圍第 5 項所述之一階延伸李德-所羅門碼之錯誤訂正裝置，更包括：

消失位置係數產生裝置，其藉由伴隨於上述收訊字組之消失旗標，產生消失位置係數；

(請先閱讀背面之注意事項再填寫本頁)

六、申請專利範圍

消失位置多項式產生裝置，其根據上述消失位置係數產生裝置所產生之消失位置係數，產生消失位置多項式；以及

修正徵候值多項式產生裝置，用以將上述徵候值多項式產生裝置所產生之徵候值多項式和上述消失位置多項式產生裝置所產生之消失位置多項式相乘，產生修正徵候值多項式；

其中上述歐氏演算法計算裝置係將上述修正徵候值多項式產生裝置所產生之修正徵候值多項式，設為初始值，並藉由歐氏演算法計算錯誤消失位置多項式和錯誤消失數值多項式；上述 Chien 搜尋裝置係根據上述歐氏演算法計算裝置所算出之錯誤消失位置多項式和錯誤消失數值多項式，計算錯誤位置和錯誤數值。

7. 一種二階延伸李德-所羅門碼之錯誤訂正裝置，其包括：

徵候值產生裝置，用以根據所輸入之收訊字組，計算出其徵候值；

收訊字組儲存裝置，用以儲存上述收訊字組；

Berlekamp-Massey 演算法計算裝置，其採用上述徵候值產生裝置所計算之徵候值，進行 Berlekamp-Massey 演算法，並藉由形式移位暫存器長度值，控制是否使用包含第二延伸成分錯誤資訊之徵候值，以依序計算錯誤位置多項式；

Chien 搜尋裝置，對於上述 Berlekamp-Massey 演算

(請先閱讀背面之注意事項再填寫本頁)

訂

六、申請專利範圍

法計算裝置所計算出之錯誤位置多項式，進行 Chien 搜尋，計算錯誤位置；

錯誤數值產生裝置，當上述 Chien 搜尋裝置所計算之錯誤位置數量適當時，利用基於上述徵候值產生裝置所計算之徵候值而產生之徵候值多項式，計算錯誤數值；以及

錯誤訂正裝置，根據上述 Chien 搜尋裝置所計算之錯誤位置和上述錯誤數值產生裝置所計算出之錯誤數值，用以訂正儲存於上述收訊字組儲存裝置中之收訊字組的錯誤。

8.如申請專利範圍第 7 項所述之二階延伸李德-所羅門碼之錯誤訂正裝置，更包括：

消失位置係數產生裝置，其藉由伴隨於上述收訊字組之消失旗標，產生消失位置係數；以及

消失位置多項式產生裝置，其根據上述消失位置係數產生裝置所產生之消失位置係數，產生消失位置多項式；

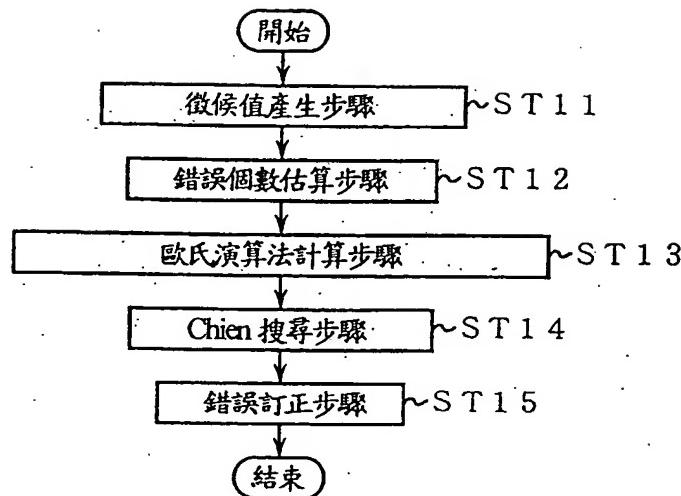
其中上述 Berlekamp-Massey 演算法計算裝置係將上述消失位置多項式產生裝置所產生之消失位置多項式，設為初始值，並且藉由形式移位暫存器長度值，控制是否使用包含第二延伸成分之錯誤資訊的徵候值，以依序計算錯誤位置多項式；上述 Chien 搜尋裝置係對於上述 Berlekamp-Massey 演算法計算裝置所算出之錯誤位置多項式，進行 Chien 搜尋。

(請先閱讀背面之注意事項再填寫本頁)

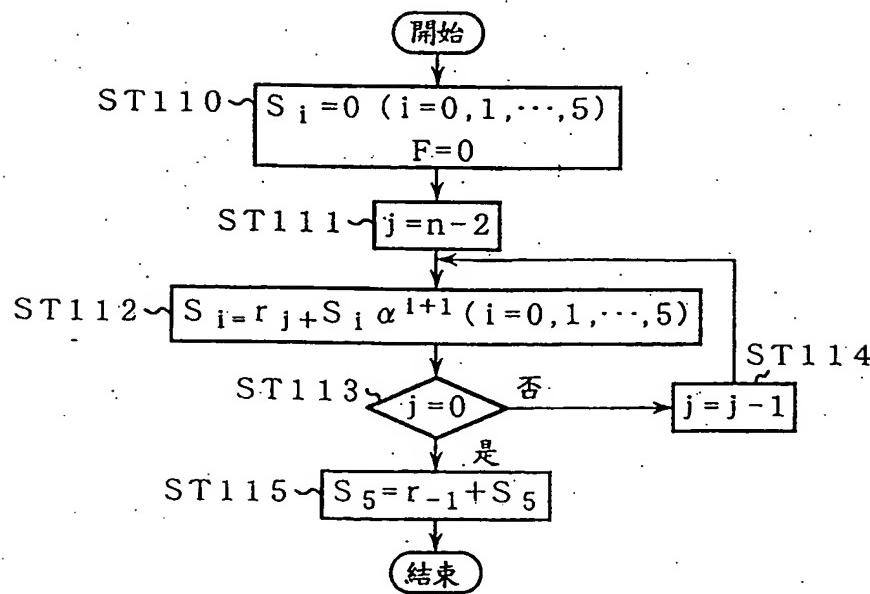
訂

A17260

87102288

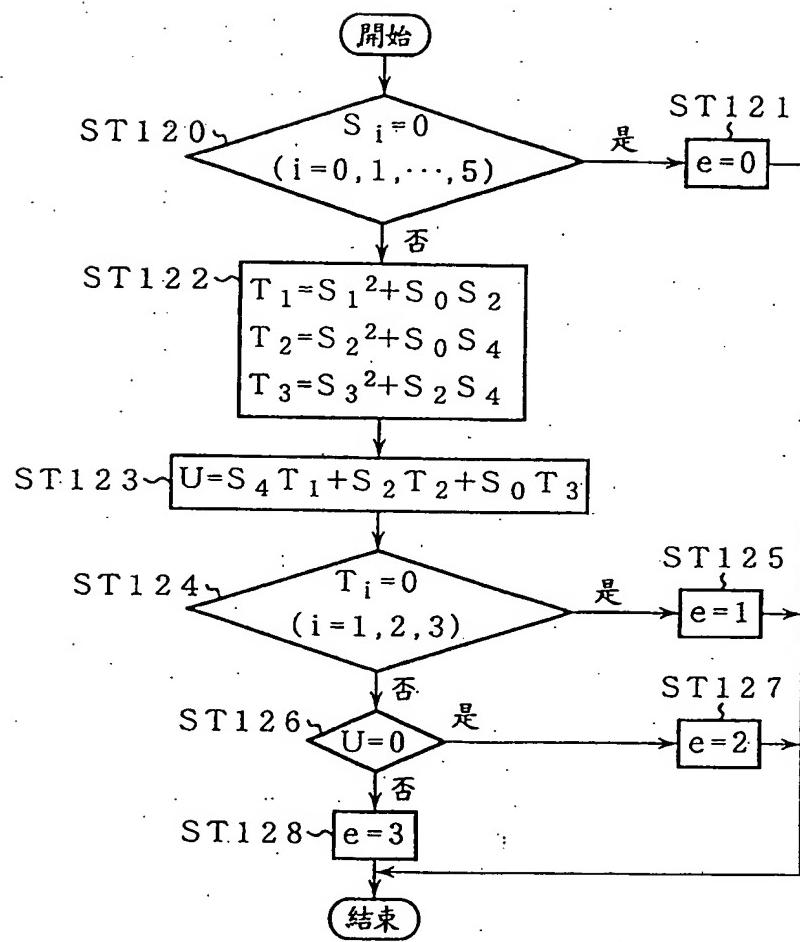


第 1 圖



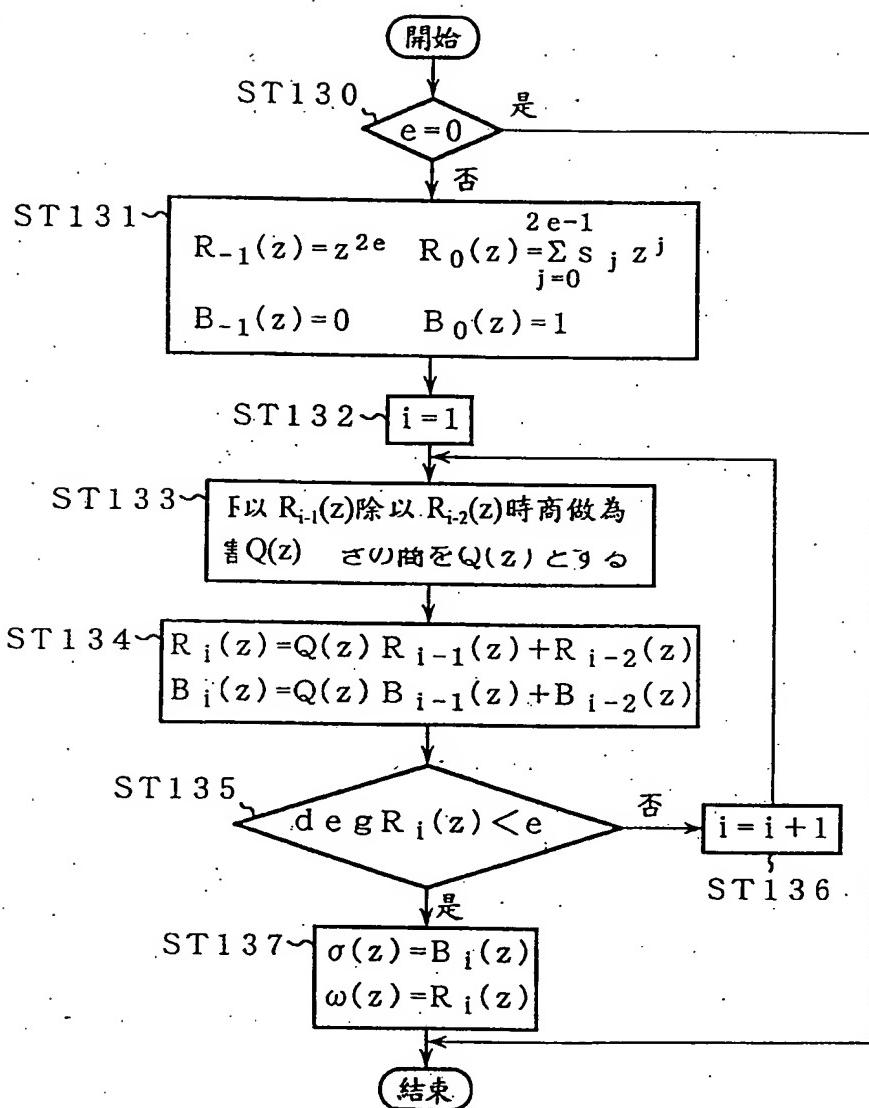
第 2 圖

417260

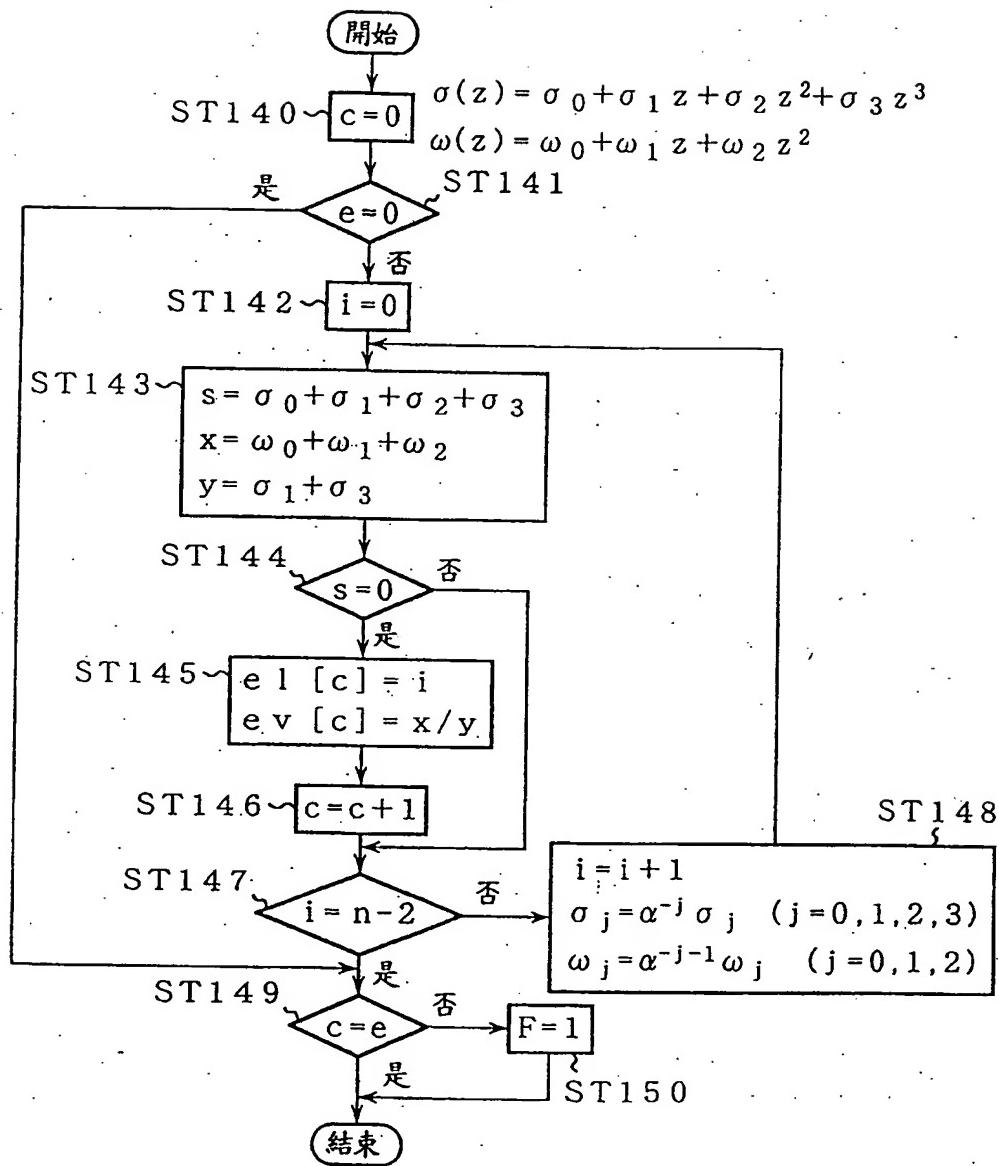


第3圖

417360

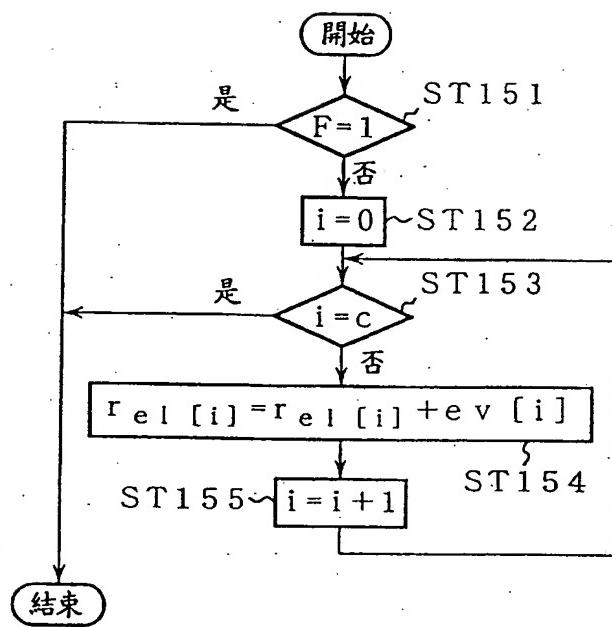


第4圖

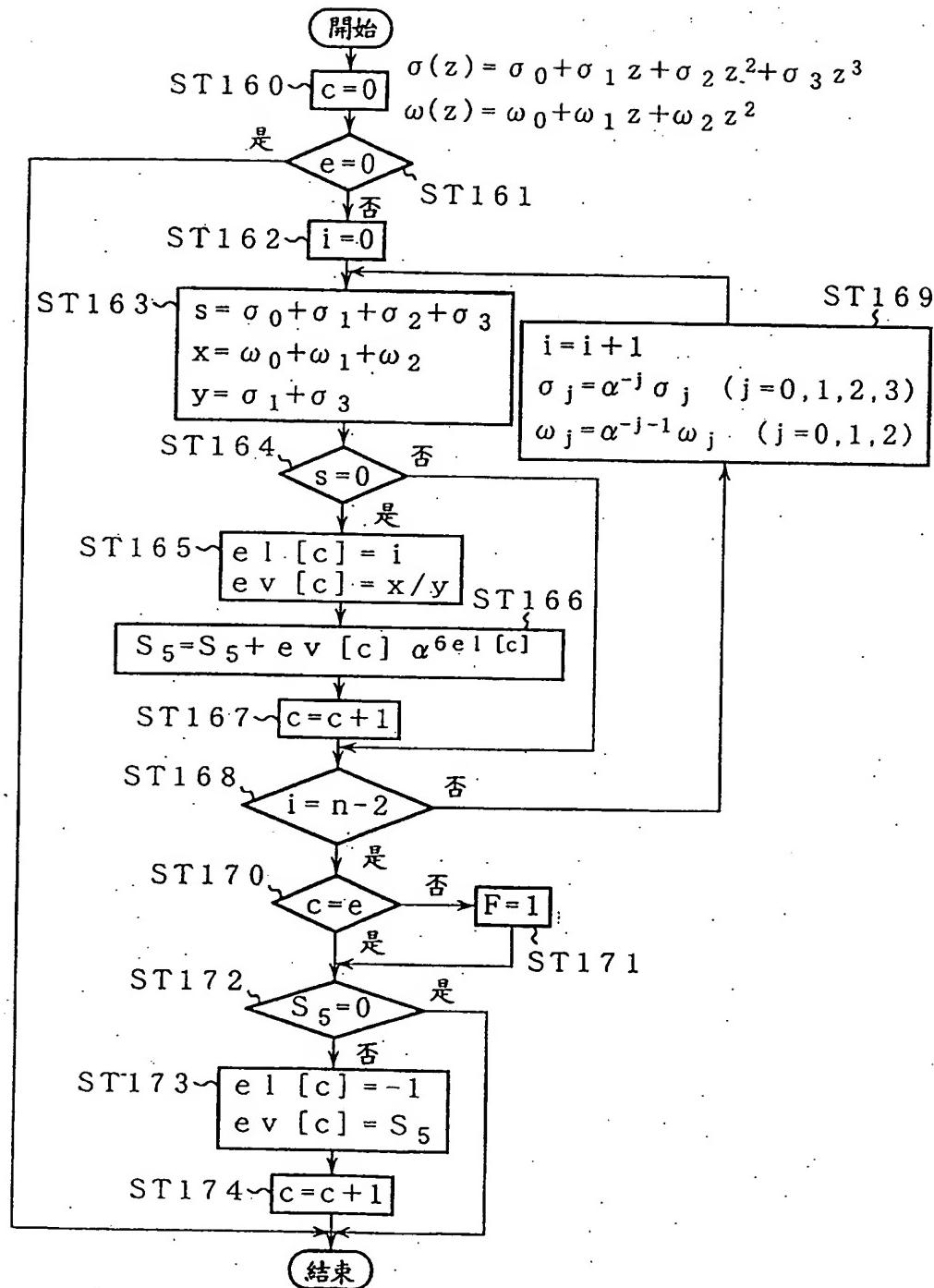


第 5 圖

417260

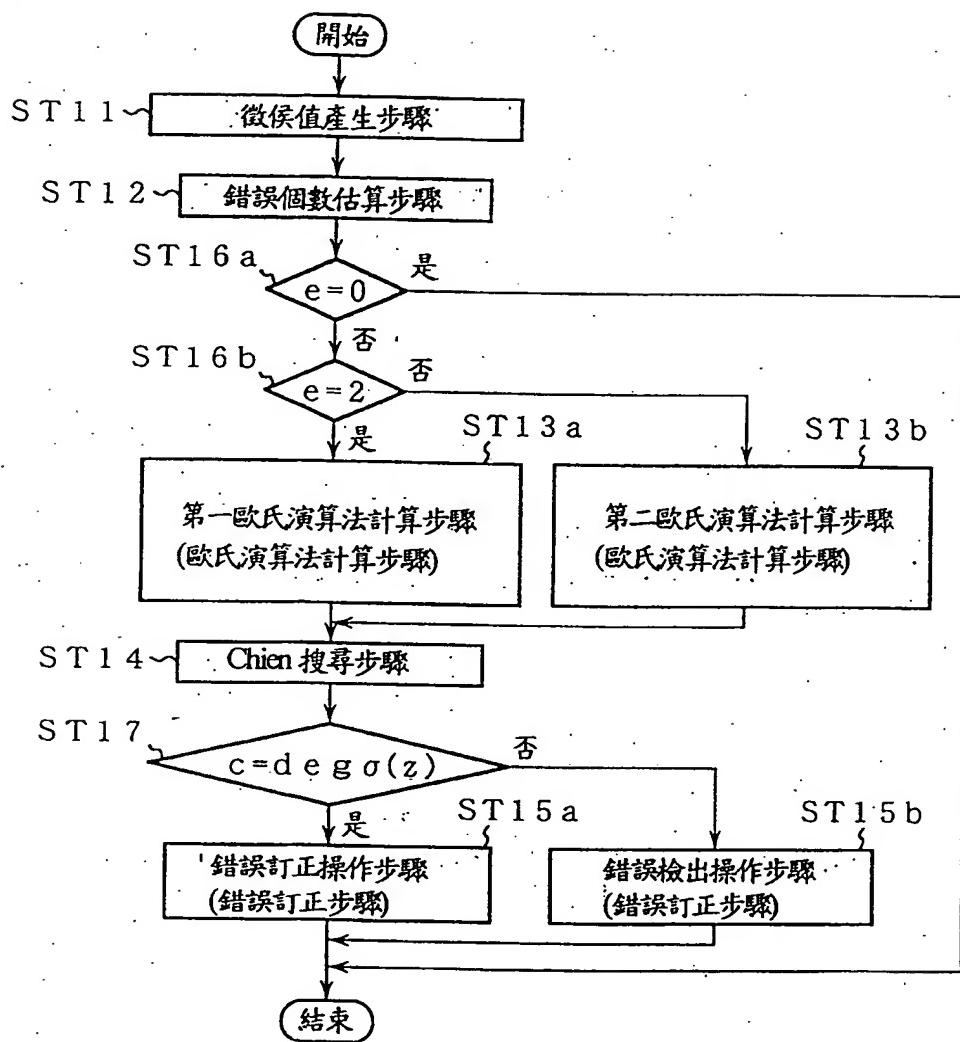


第 6 圖



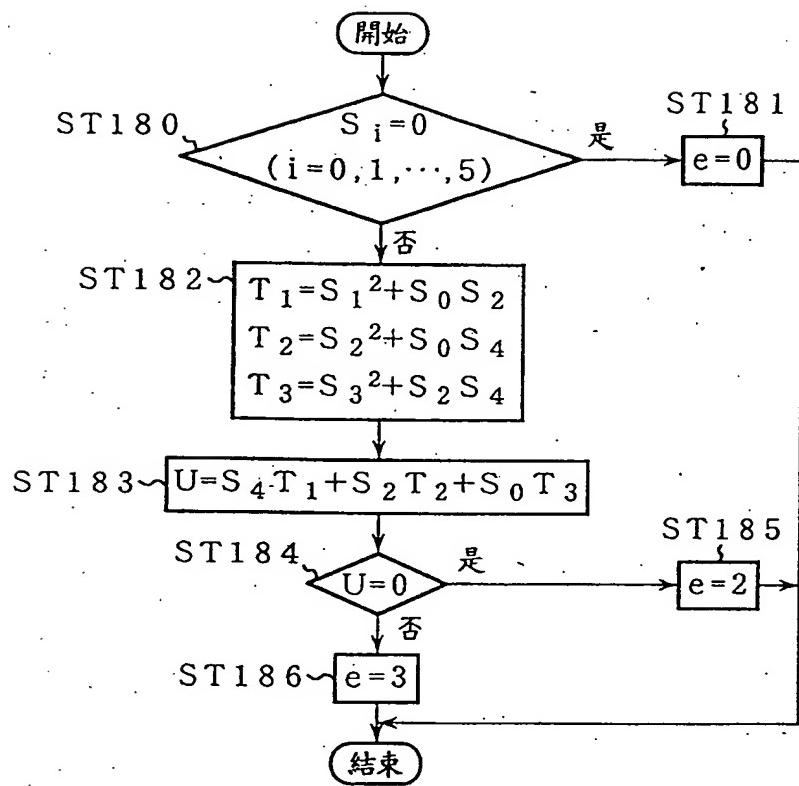
第 7 圖

417360

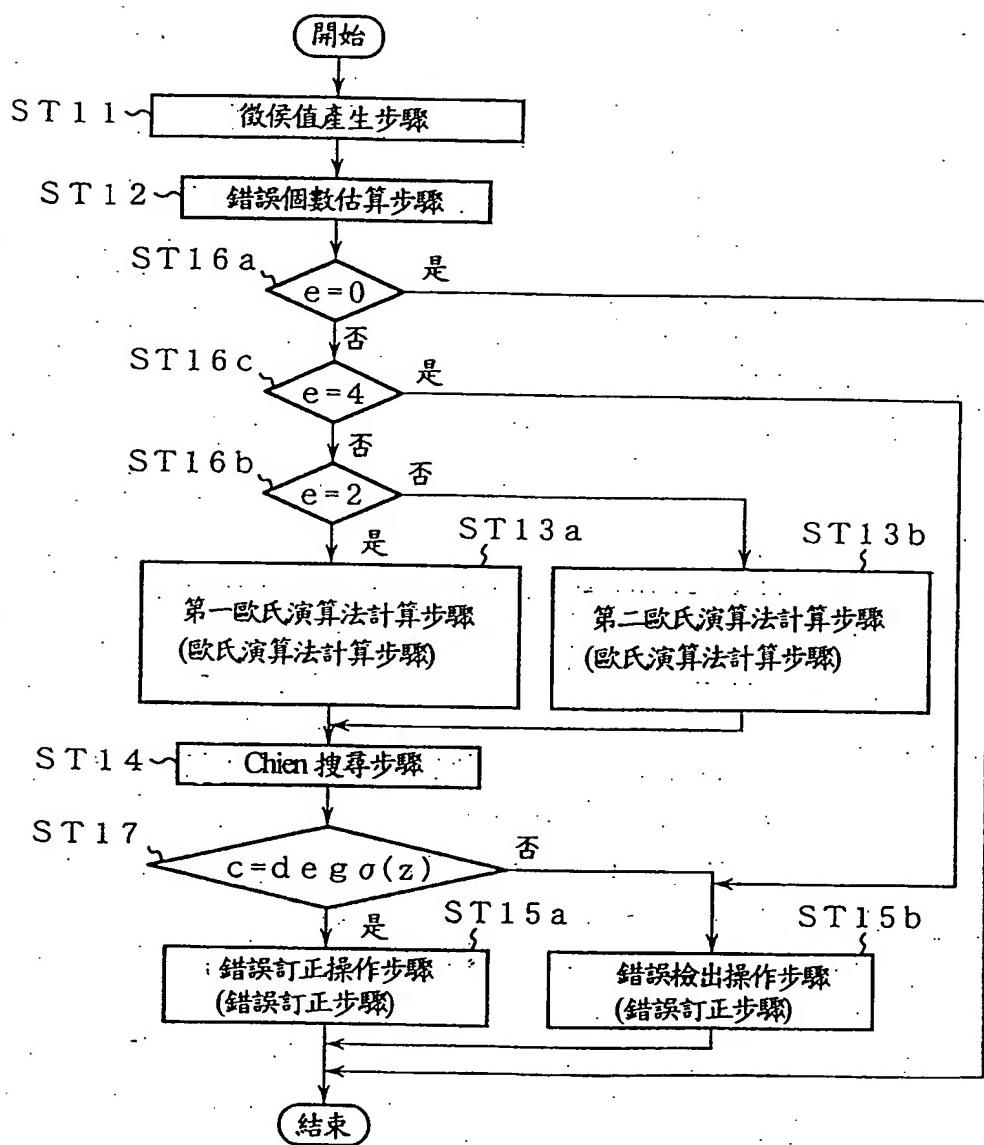


第 8 圖

417360

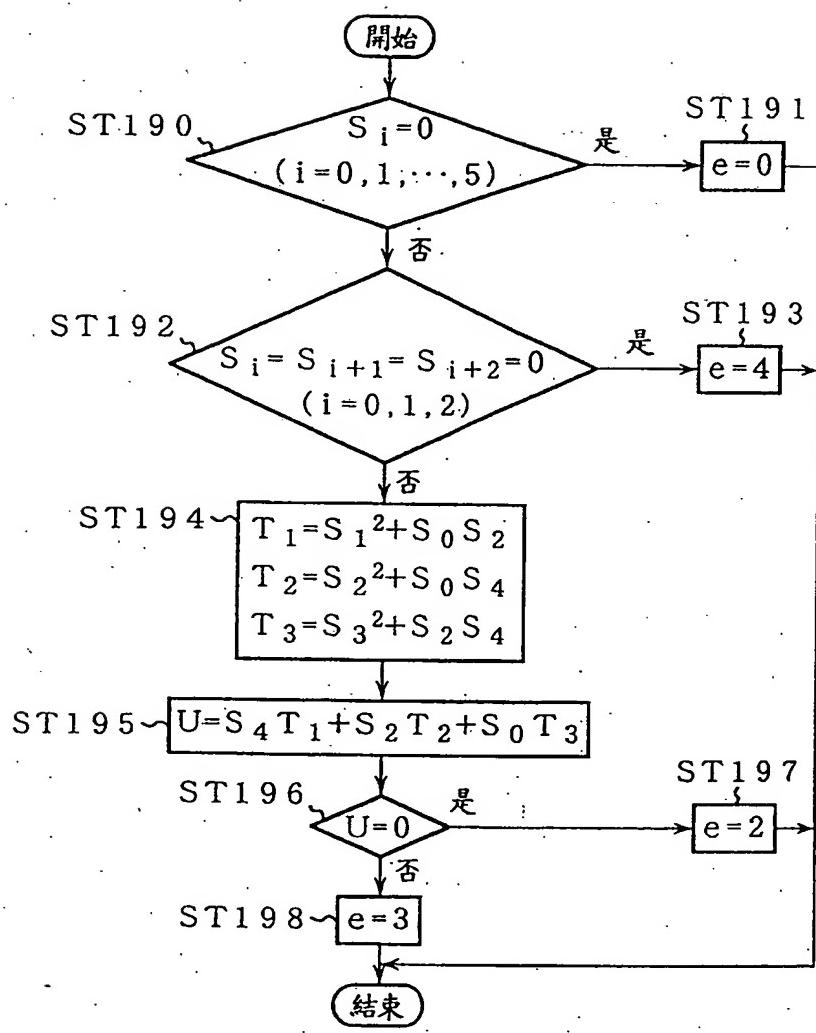


第9圖



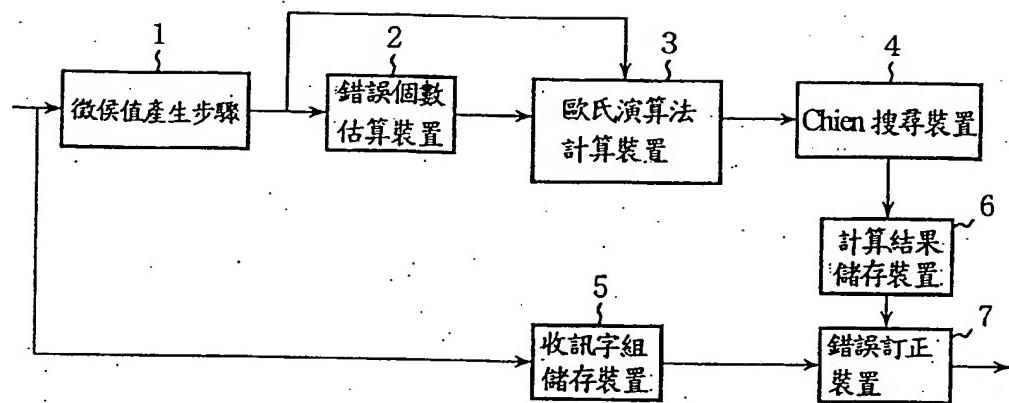
第 10 圖

417360

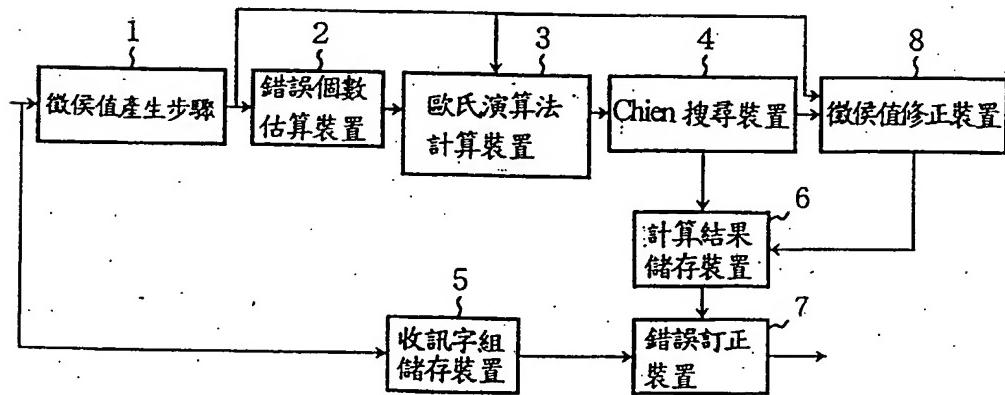


第 11 圖

417360

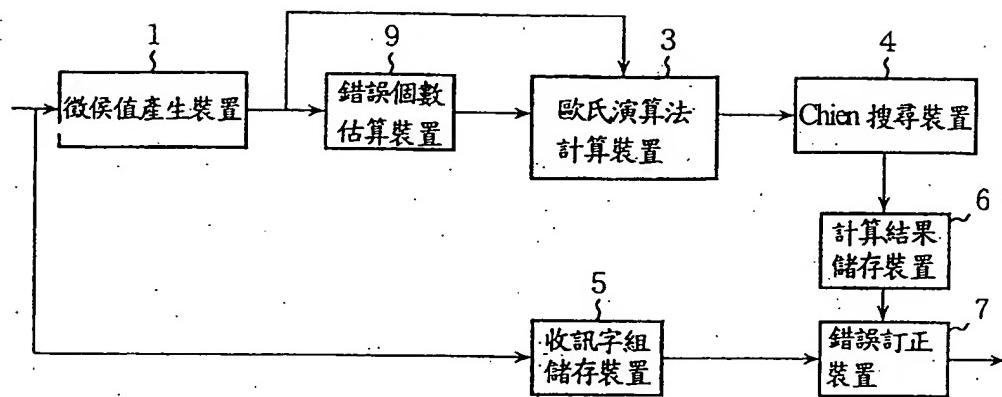


第 12 圖

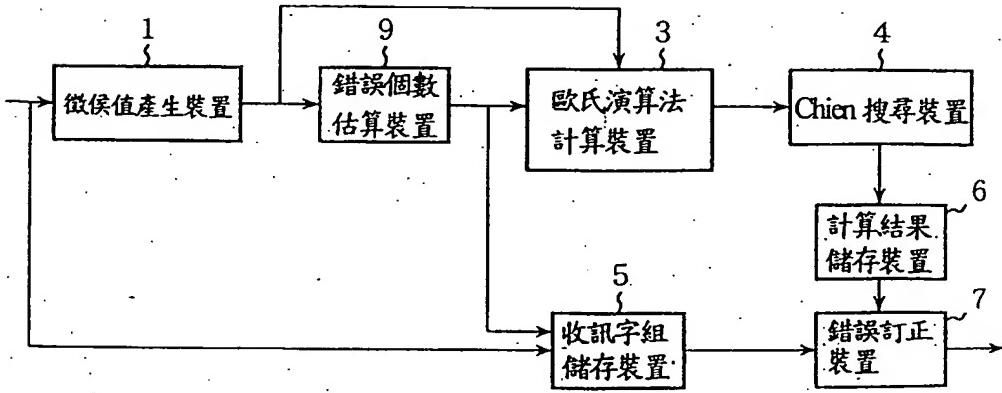


第 13 圖

417360

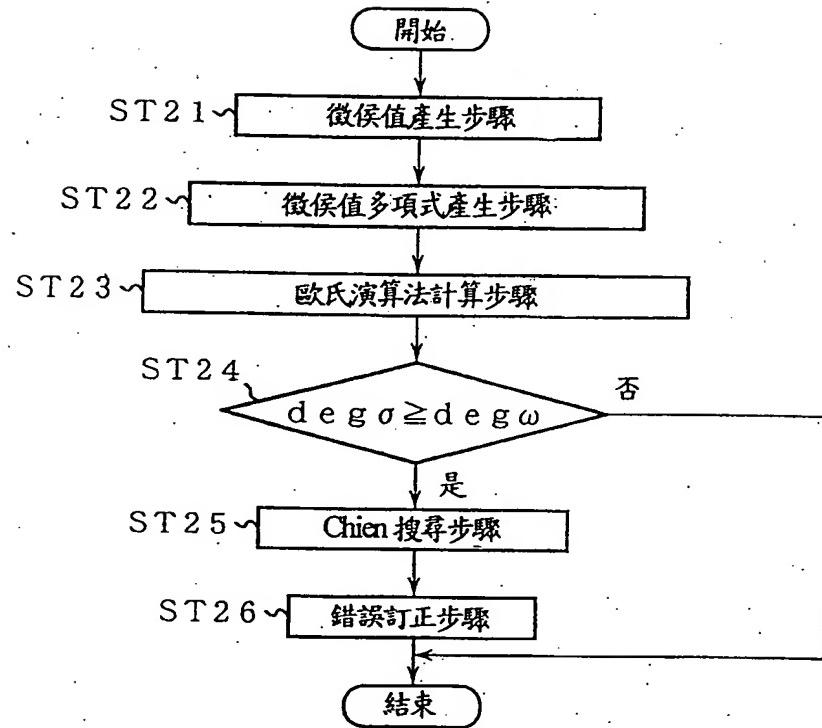


第 14 圖

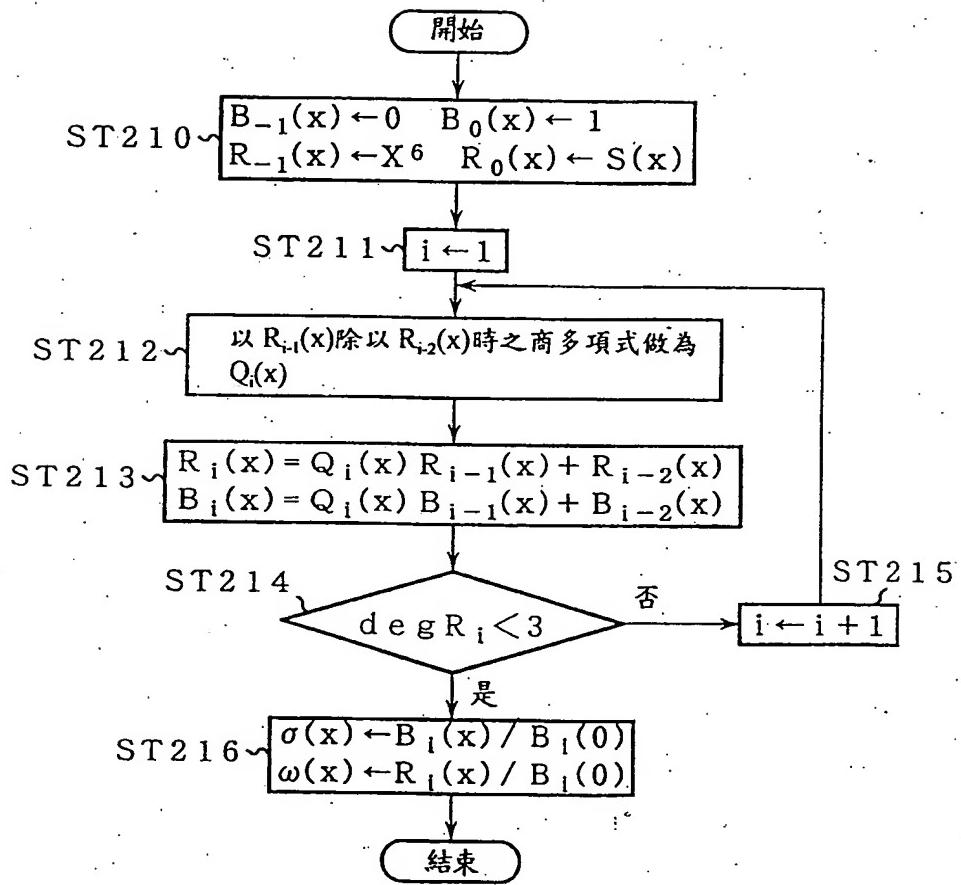


第 15 圖

417360

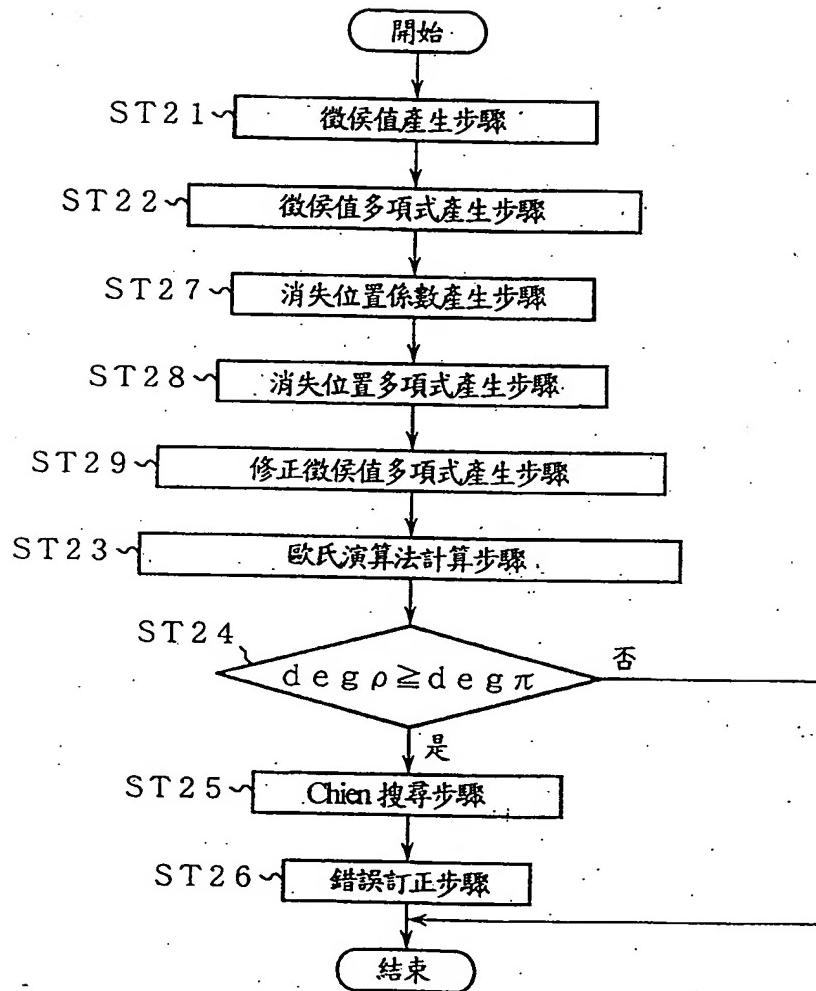


第 16 圖

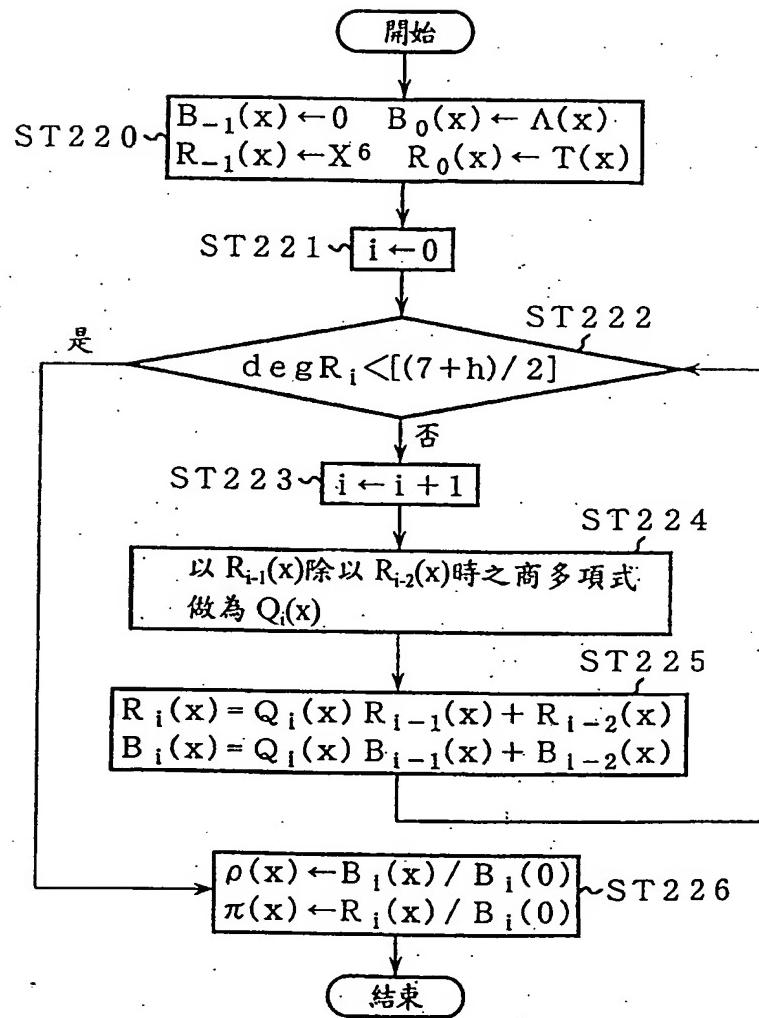


第 17 圖

417269

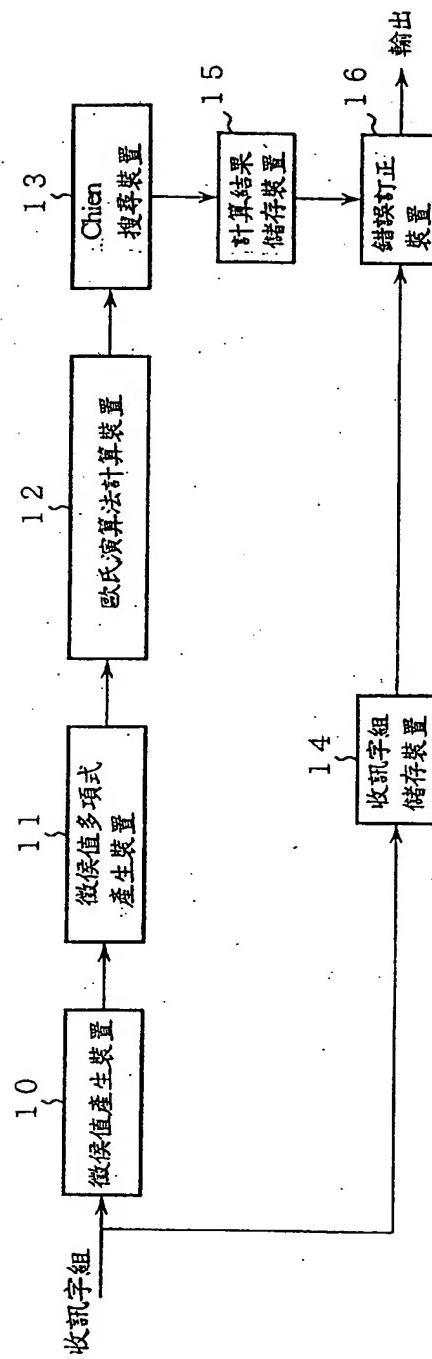


第 18 圖

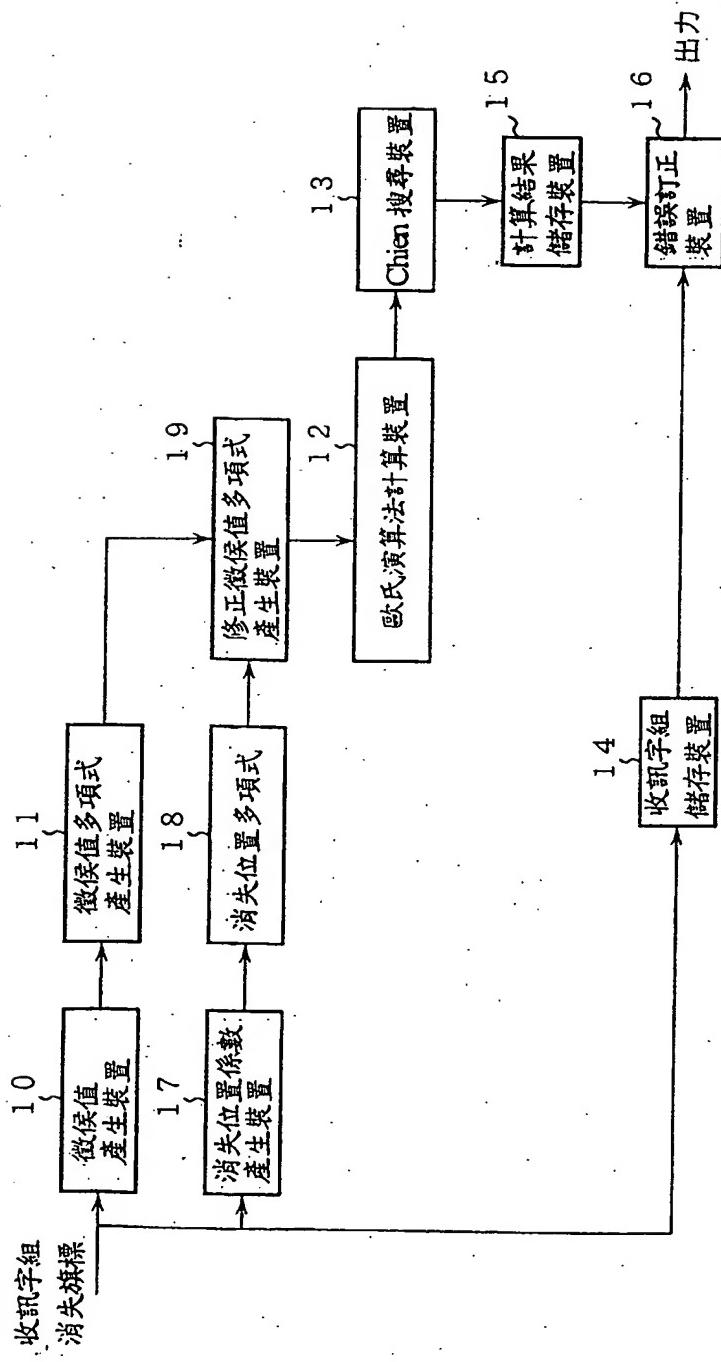


第 19 圖

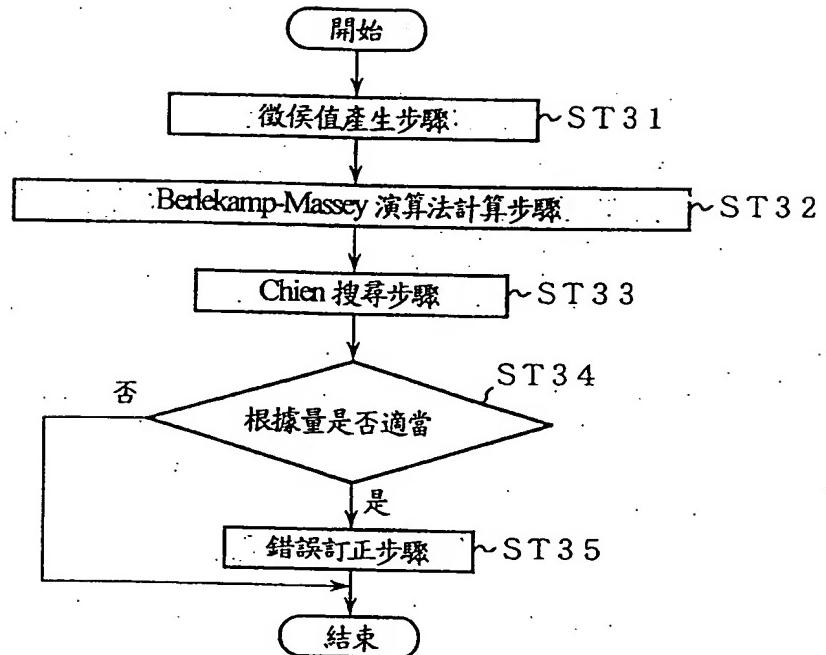
第 20 圖



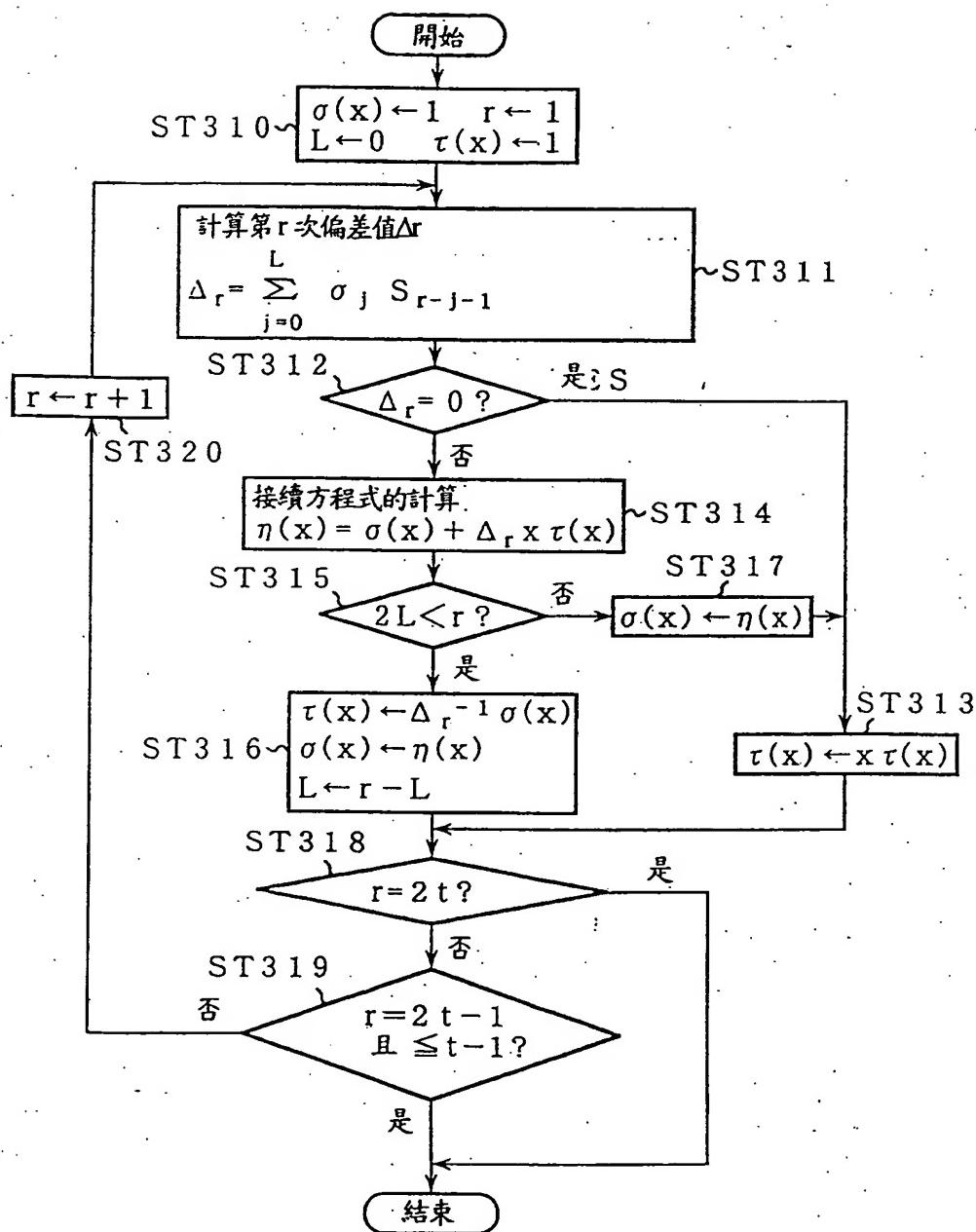
第 21 圖



417260

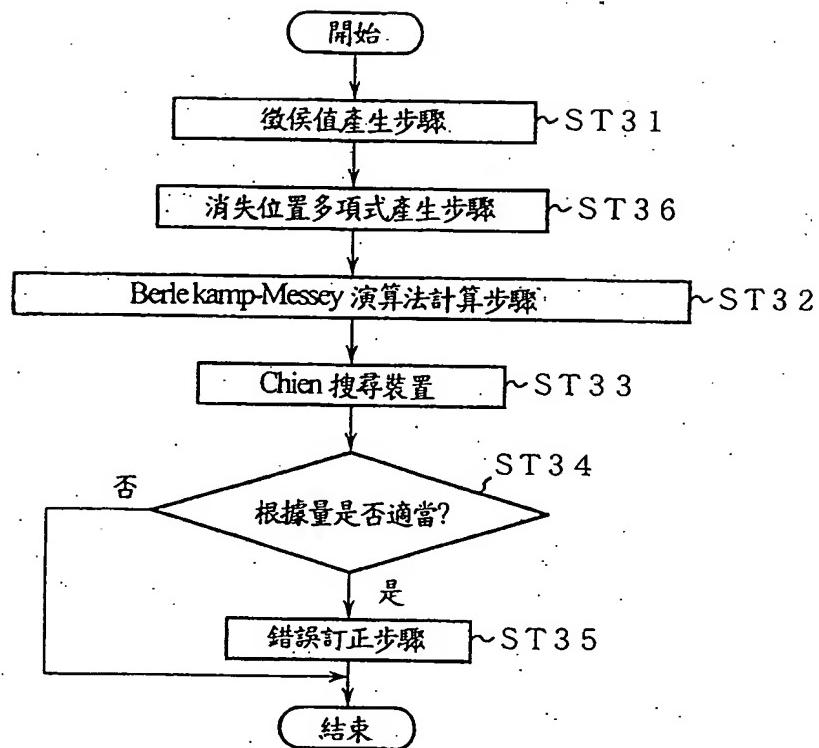


第 22 圖

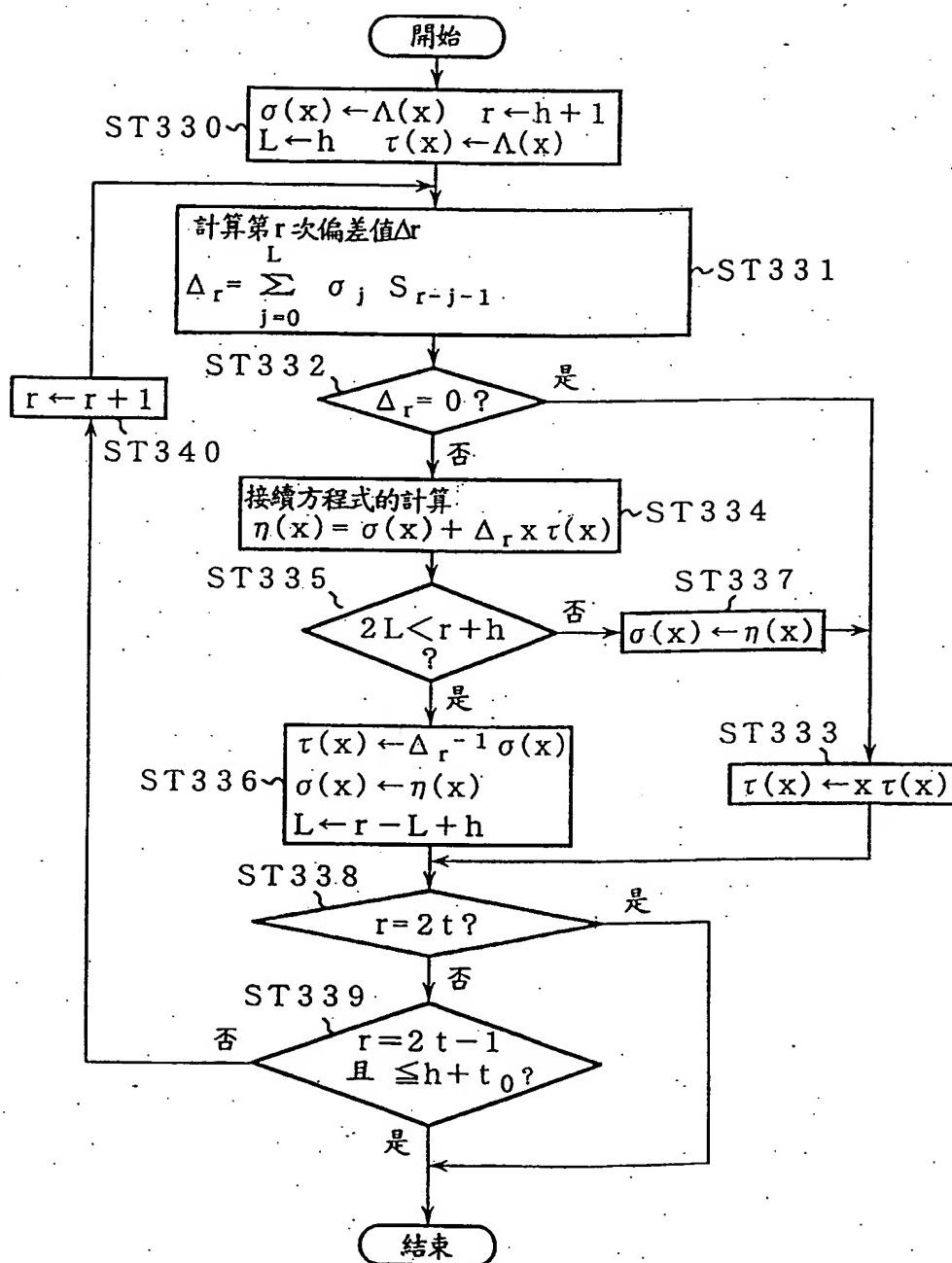


第 23 圖

417360

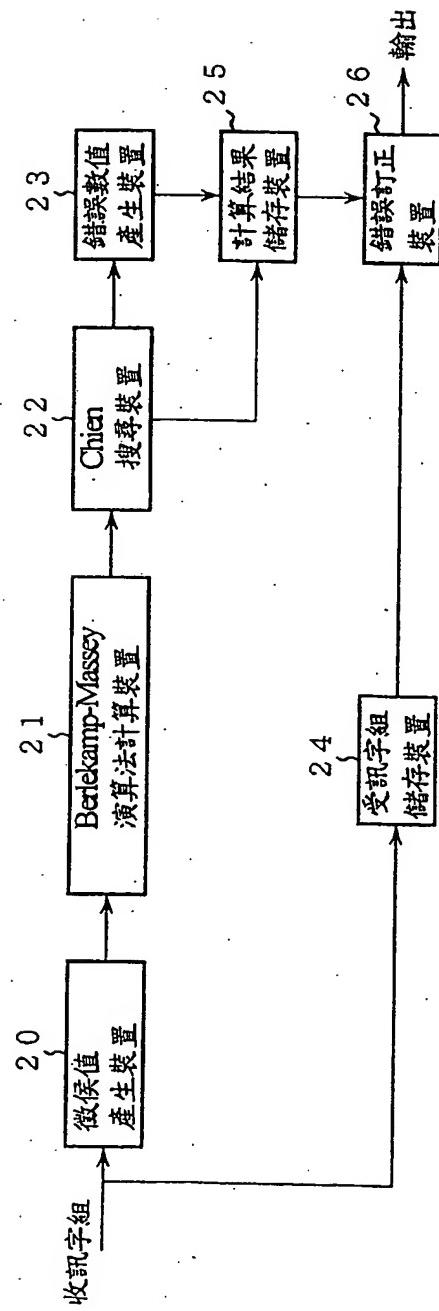


第 24 圖



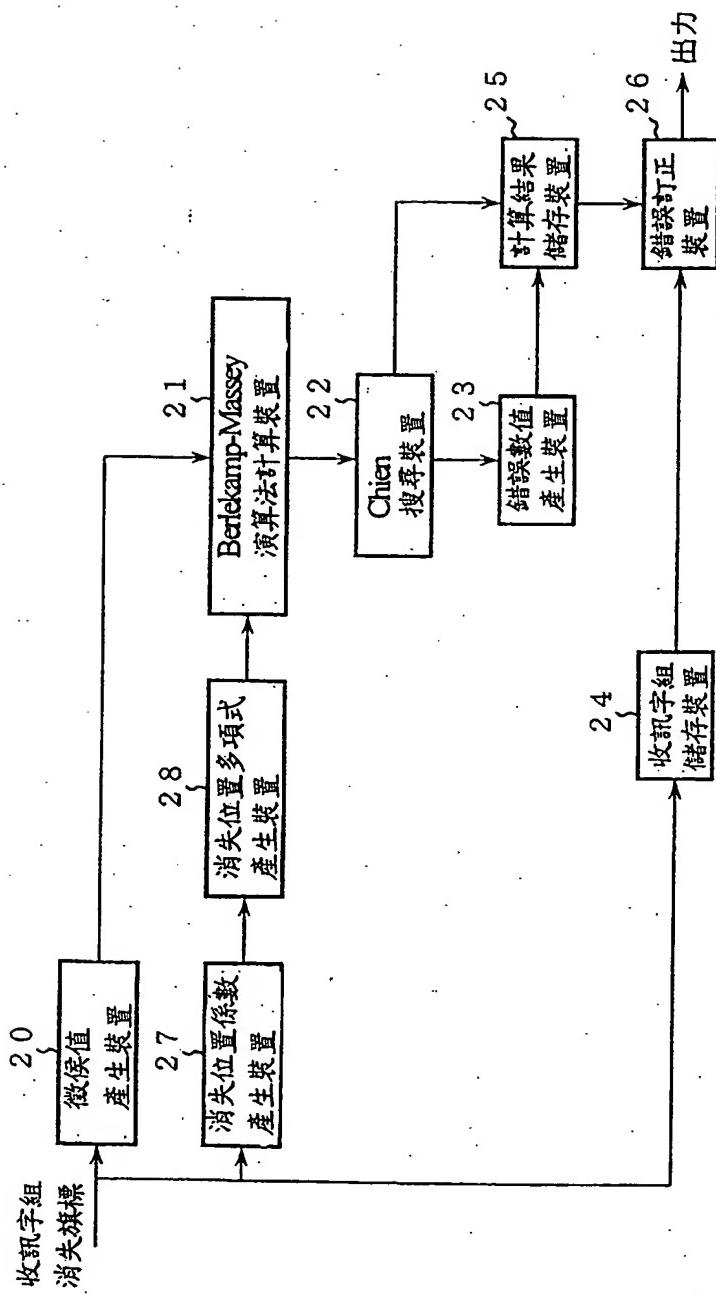
第 25 圖

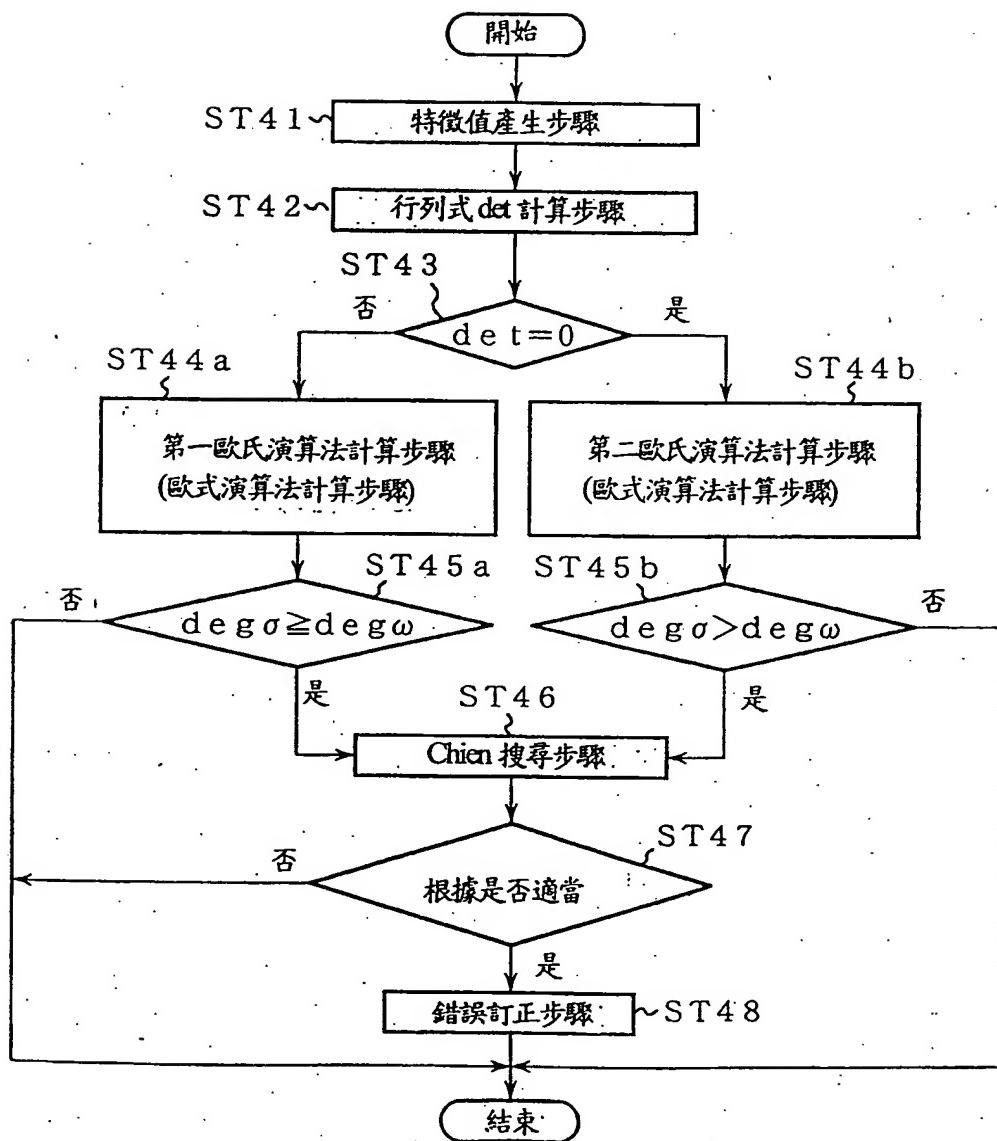
第 26 圖



417360

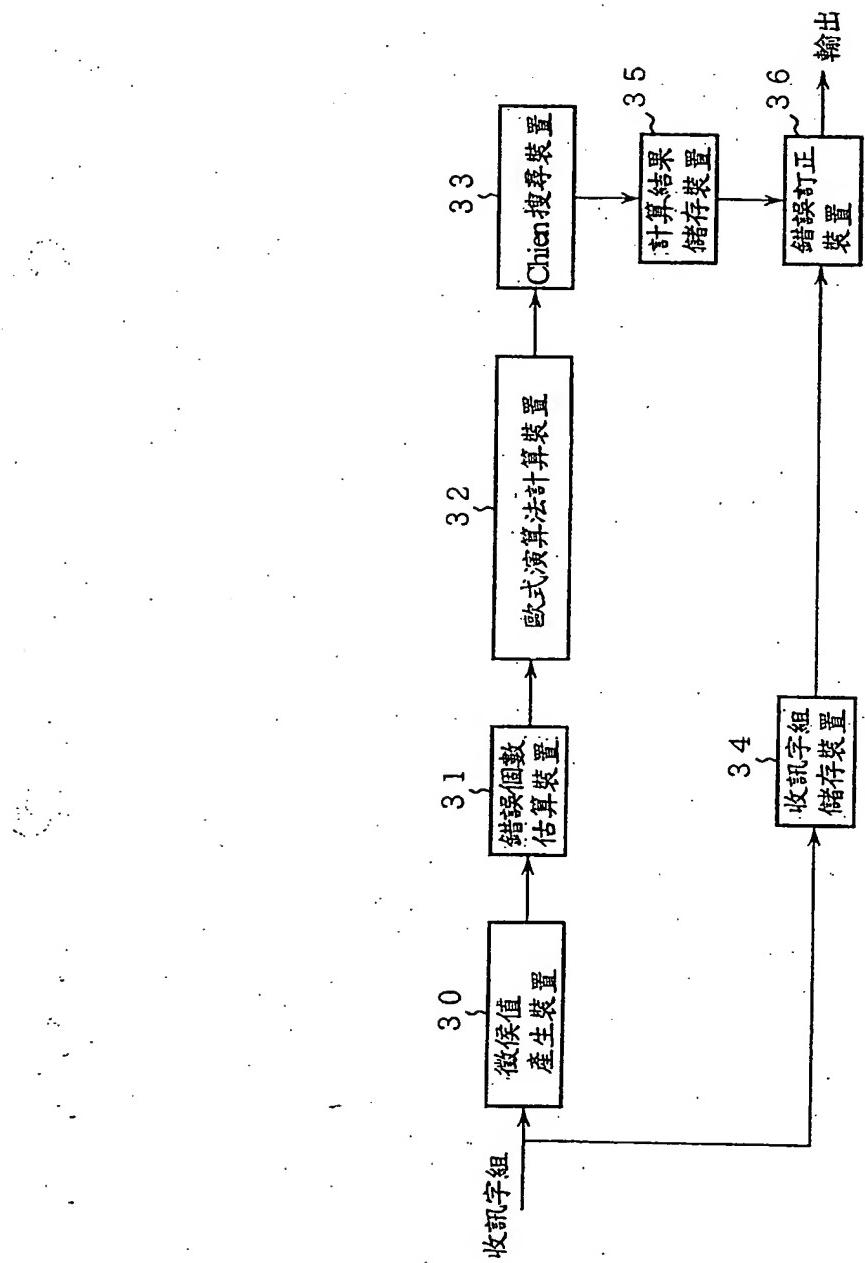
第 27 圖





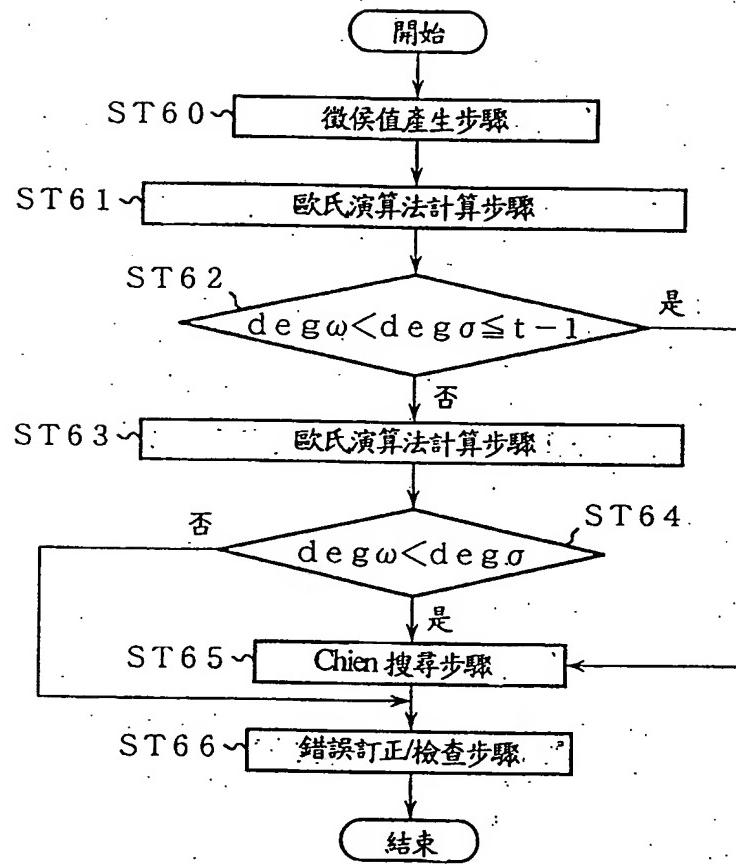
第 28 圖

第 29 圖



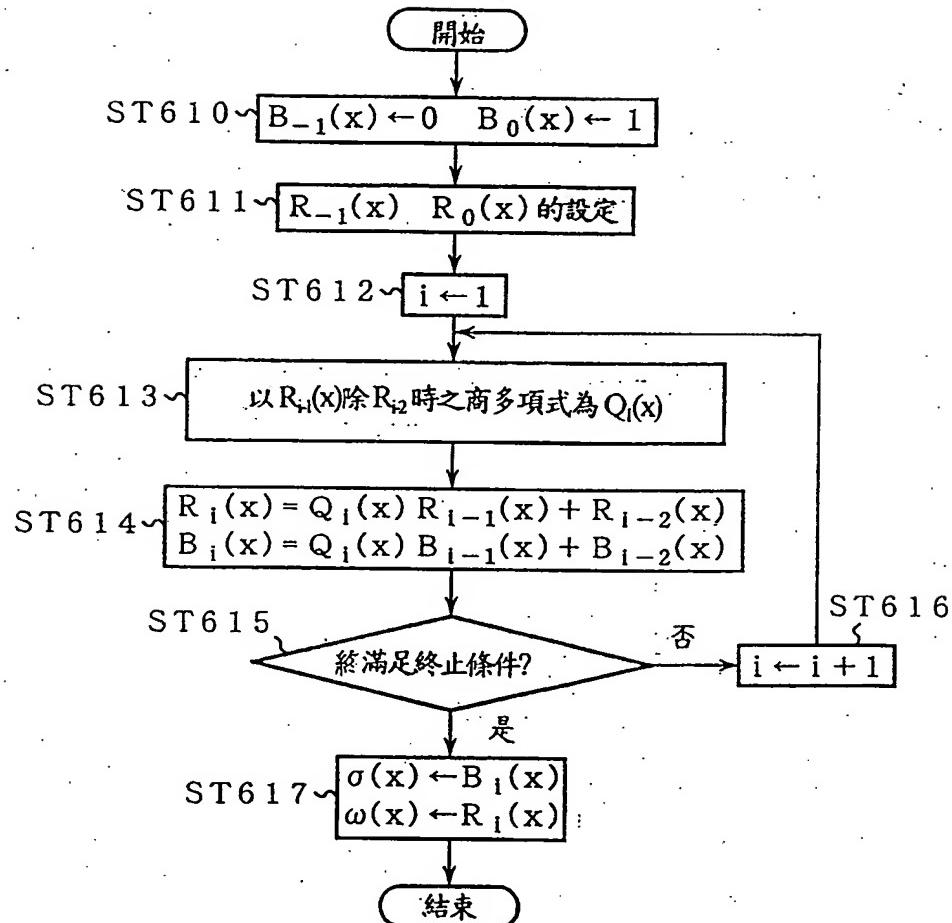
41766

417360

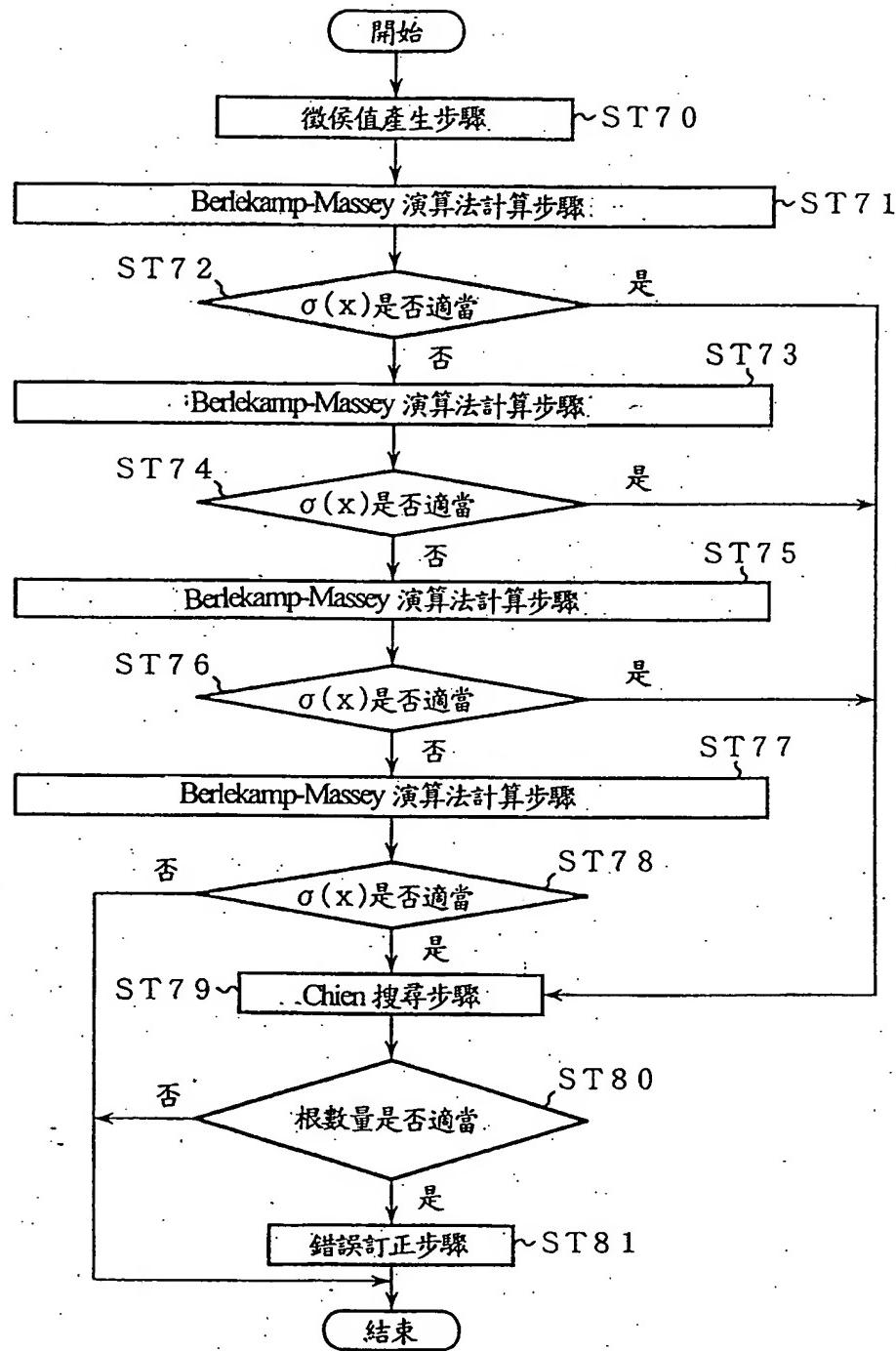


第 30 圖

417369

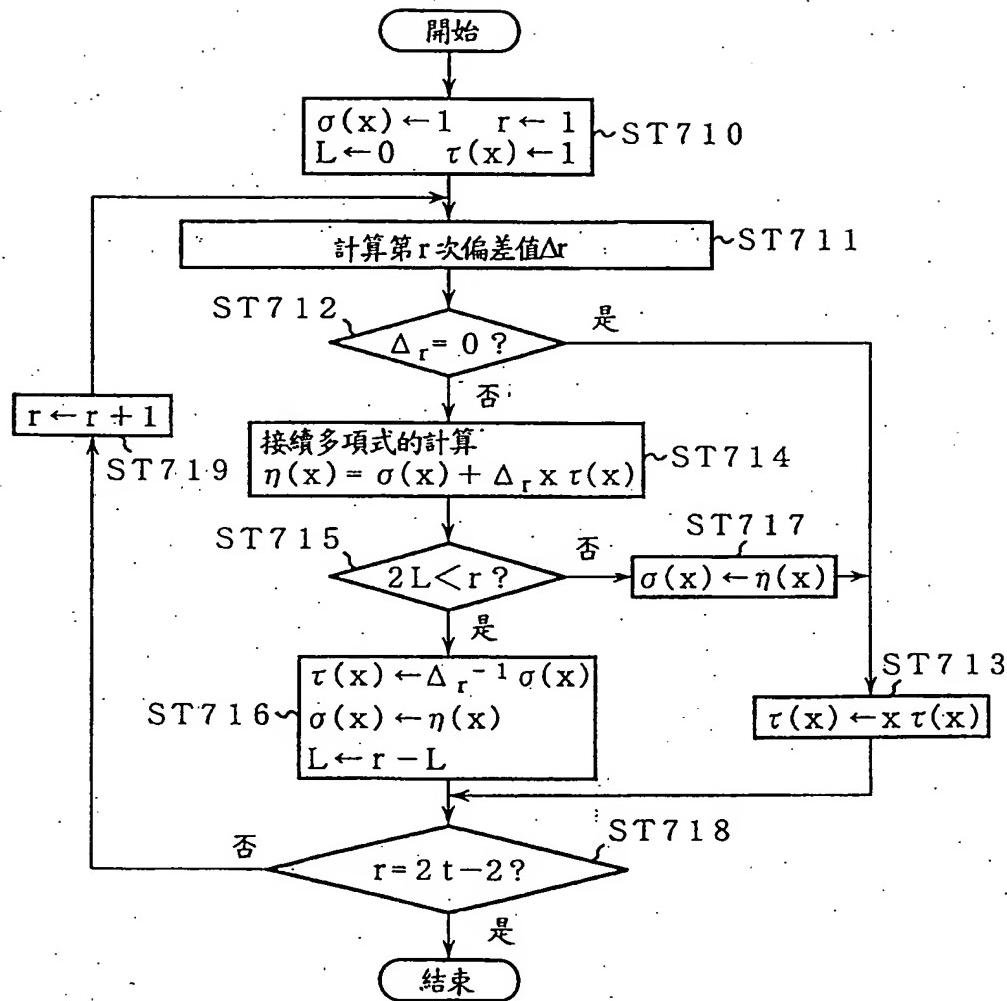


第 31 圖



第 32 圖

417360



第 33 圖